

**Towards an Over-the-horizon Awareness to Driver Support
Systems in Highway Real-World Scenarios**

Master's Thesis

to obtain the title of

Master of Science in Telematics

Defended by

Ramon S. Schwartz

Chair for Design and Analysis of Communication Systems (DACS)
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
University of Twente, The Netherlands

Thesis Advisors:

Martijn van Eenennaam, M.Sc.
Dr. ir. Georgios Karagiannis
Dr. ir. Geert J. Heijenk
Wouter Klein Wolterink, M.Sc.

August 24th, 2009



Supported by the Programme ALβan
The European Union Programme of High Level Scholarships for Latin America.
Scholarship Number: E07M401331BR

Abstract:

Aiming at addressing the high level of congestion due to the increasing number of vehicles on roads worldwide, Driver Support Systems (DSS) have been proposed to assist drivers on the road by improving safety, efficiency and comfort in the driving experience. One important benefit of deploying such systems is that by aiding drivers in traversing traffic congestion on highways and consequently having vehicles moving more smoothly close to a constant speed, the emission of CO_2 in the environment is reduced [Johannson 1999].

In the same context, a novel system referred to as the Congestion Assistant has been proposed in [van Driel 2007] by C.J.G. van Driel. In this work, a user needs survey conducted to investigate the perceived needs for driver assistance indicated high user acceptance, in particular, in receiving assistance with the driving task on congested roads. Based on user preferences, the Congestion Assistant has been designed and later evaluated by means of traffic simulation. The assessment of the system focused on traffic efficiency and safety and demonstrated reduction in congestion for all variants of Congestion Assistant considered. Nevertheless, some of its basic operations require information regarding the upcoming traffic condition on the road and no system to acquire it has been proposed.

Given the advantage of rapid dissemination of traffic information by means of vehicular communication and the necessity for the Congestion Assistant to acquire information regarding the upcoming traffic, Martijn van Eenennaam has introduced in [van Eenennaam 2008] an efficient means to provide an *over-the-horizon awareness* of traffic jams ahead on the road to vehicles by means of multi-hop vehicle-to-vehicle communication. The proposed solution is a communication protocol that complies with information requirements derived from the Congestion Assistant. Nevertheless, this solution has been designed with one-dimensional straight highway scenarios in mind and extensions and/or adaptations might be necessary to make it solution adequate for more real-world scenarios.

In this work, we present a networking solution, the Over-the-Horizon Awareness (OTHA) protocol, that altogether comprises extensions and modifications to the communication protocol presented in [van Eenennaam 2008] in order to address more realistic scenarios. In particular, the following Highway Real-World Scenarios are considered: *single-lane roads*, *multiple-lane roads*, *junctions*, and *roads with multiple (opposite) directions*.

The performance of the OTHA protocol is evaluated by means of simulation. We assess the protocol both in a controlled environment with static scenarios and under a more realistic scenario consisting of vehicle traces with high mobility and speed variations. The results obtained indicate good performance of the protocol with respect to the metrics evaluated. In particular, the performance of the protocol in multiple-lane scenarios is found to deteriorate in high vehicle densities and possible solutions to overcome this problem are described.

Keywords: Road Traffic, Congestion, Vehicle-to-vehicle, Communication Protocol

Acknowledgments

I would like to start by thanking Martijn van Eenennaam and Georgios Karagiannis for all their advise and unconditional help during the development of this work. I also would like to express my gratitude to Geert J. Heijenk and Wouter Klein Wolterink, who provided me with great insight in the organization and ideas for this work. You all have taught me a lot during these months and I am very grateful for that.

I am also thankful to all people that supported me during these two years in The Netherlands. A special thank you to my girlfriend Rita, without whom I would probably not obtain all the energy I had in the final steps of my work.

Finally but not less important was the complete support I had from my family back in Brazil: my mother (Clenice), my father (Liberato), and my brothers (Sissy and René).

Contents

1	Introduction	1
1.1	Research Objectives	3
1.1.1	Research questions	3
1.2	Approach	3
1.3	Outline	4
2	Background Information: the IEEE 802.11p standard	5
2.1	A brief overview on IEEE 802.11	5
2.1.1	Challenges	7
2.2	802.11 in vehicular communication	8
3	The Congestion Assistant Requirements	11
3.1	System Overview	11
3.2	Requirements for Real-World Scenarios	14
3.2.1	Multi-lane highways	14
3.2.2	Roads with different directions	15
3.2.3	Junctions	17
3.3	System Requirements	19
3.4	The Navigation System	21
3.4.1	Positioning sensor	21
3.4.2	Speed sensor	22
3.4.3	Headway sensor	22
3.4.4	Lane sensor	22
3.4.5	Geographical map	23
4	The Over-the-Horizon Awareness Protocol Overview	25
4.1	The Protocol Overview	25
4.2	Communication Requirements and Assumptions	28
4.2.1	Assumptions	28
4.2.2	Communication Requirements	28
5	The Traffic Filter Protocol Layer	31
5.1	The TrafficMap	31
5.1.1	The TrafficMap structure	34
5.2	Traffic Filtering	35
5.3	The Traffic Filter Protocol Layer	39
5.3.1	Merging information	42
5.3.2	Adding an entry	45
5.3.3	Averaging	46
5.3.4	Reducing the TrafficMap	47

5.4	Compression Optimization	50
5.4.1	Greedy approach	52
5.4.2	Follow-lane approach	53
5.4.3	Analysis and comparison between the approaches	57
5.5	Summary of Parameters	61
6	The Dissemination Protocol Layer	63
6.1	Background	63
6.2	Addressing Source Vehicles	67
6.3	The Dissemination Protocol Layer	69
6.3.1	Time Manager	71
6.3.2	Message Builder	78
6.4	Time Slot Optimization	83
6.5	Summary of Parameters	85
7	Performance Evaluation of the OTHA Protocol	87
7.1	Evaluation Metrics	87
7.1.1	General metrics	88
7.1.2	Mobility-specific metrics	92
7.1.3	Compression-specific metrics	92
7.2	Simulation Configuration	93
7.3	Static Scenarios	96
7.3.1	Scenario description	96
7.3.2	Results	98
7.4	Mobility Scenarios	108
7.4.1	Scenario description	108
7.4.2	Results	113
7.5	Optimization Methods	118
7.5.1	Compression	118
7.5.2	Time Slot	120
8	Conclusion	123
8.1	General Conclusions	123
8.2	Answers to Research Questions	124
8.3	Future Work	126
A	Illustration of Static Scenarios	129
B	TrafficMap Entries	137
B.1	Static Scenario	137
B.1.1	Single-lane	137
B.1.2	Multiple-lane	137
B.1.3	Junction	138
B.1.4	Opposite direction	138
B.2	Mobility Scenario	138

List of Figures

2.1	IEEE 802.11's Basic Access Method [Schiller 2003]	7
2.2	The Hidden Terminal Problem	8
2.3	DSRC spectrum band and channels in the US	9
3.1	Abstract of Intelligent Transport Systems within a vehicle	12
3.2	Overview of the Congestion Assistant functioning [van Eenennaam 2008]	13
3.3	An illustration and a real example of multi-lane highways	16
3.4	Example of a spaghetti junction [Cozart 2000]	17
3.5	Example of a junction with entrance and exit points	18
3.6	Abstraction of junctions in highways	19
3.7	Interactions between the Congestion System and its required applications	20
4.1	Basic functioning of the OTHA protocol	26
4.2	Overview of the Over-the-Horizon Awareness (OTHA) Protocol	27
5.1	Separate TrafficMaps with the following notation: L (lane), E (entry), P (position) and S (speed)	32
5.2	Merged TrafficMaps with the following notation: L (lane), E (entry), P (position) and S (speed)	33
5.3	The TrafficMap Structure	34
5.4	The threshold-based approach for a road with a single lane	37
5.5	The threshold-based approach for a road with multiple lanes	38
5.6	Top view representation of the road for different traffic conditions	38
5.7	Example of vehicle classification	39
5.8	Overview of The Traffic Filter Protocol Layer	40
5.9	The Prepare TrafficMap Information Process	43
5.10	Example of TrafficMap information being merged	44
5.11	The threshold-based <i>sensitivity</i> ε function	46
5.12	Weight θ as a function of distance d for different parameters of Δ and α	48
5.13	Example of reducing the TrafficMap based on the current situation of roads ahead	49
5.14	Comparison between no compression and the Greedy approach	54
5.15	Example of the Follow-lane approach being employed	57
5.16	Illustration of the difference between both compression approaches	59
5.17	Exemplification of the gap problem in an execution of the Follow-lane algorithm	60
6.1	Illustration of the three broadcast suppression techniques proposed in [Wisitpongphan <i>et al.</i> 2007]	66
6.2	Overview of the dissemination protocol proposed	68
6.3	The Dissemination Protocol Layer	70
6.4	Illustration of the time slot assignment with two slots for source vehicles and three for relay vehicles	75

6.5	The Time Manager	77
6.6	The unique source ID problem	79
6.7	The problem of simply canceling messages just based on the type of vehicles. The information contained in msg ID2 is lost because vehicle C_3 cancels its transmission after hearing msg ID4.	80
6.8	A solution for the source ID problem	81
6.9	The message structure	82
6.10	The Message Builder	83
7.1	The sampling error calculation method	91
7.2	Illustration of the static scenarios considered	97
7.3	Reachability x density	99
7.4	Delay x density	100
7.5	Number of receptions x density	101
7.6	Number of transmissions x density	102
7.7	Overhead x density	102
7.8	Channel utilization x density	103
7.9	Slot utilization x density	105
7.10	Accuracy x density	106
7.11	Illustration of the matching of TrafficMap entries and the real vehicle trace for one the lanes of the multiple-lane road scenario for a density of 20 vehicles/km/lane	107
7.12	Illustration of the mobility scenario considered	109
7.13	Illustration of the mobility scenario: density x time	110
7.14	Illustration of the mobility scenario: speed x position (Section 1 -> 2)	111
7.15	Illustration of the mobility scenario: speed x position (Section 3)	111
7.16	Illustration of the mobility scenario: speed x position (Section 4)	112
7.17	Maximum theoretical distance of awareness x time	113
7.18	Maximum distance of awareness achieved x time	114
7.19	Boxplot illustrating the accuracy	115
7.20	Illustration of the matching of TrafficMap entries and the real vehicle trace for lane 1 of Section 1	116
7.21	Evaluation of the system load	117
7.22	Percentage of compression x density	119
7.23	TrafficMap size x density	119
7.24	Accuracy x density	120
7.25	Evaluation of the time slot optimization method	121
A.1	Single-lane	129
A.2	Multiple-lane - Lane 1	130
A.3	Multiple-lane - Lane 2	131
A.4	Junction - Road 1	132
A.5	Junction - Road 2	133
A.6	Opposite direction - Direction 1	134
A.7	Opposite direction - Direction 2	135

List of Tables

3.1	Classification of the required information	14
3.2	Required information considering the new scenarios described	20
5.1	Comparison between the compression approaches	60
5.2	Parameters defined in the Traffic Filter Protocol Layer	61
6.1	Parameters defined in the Dissemination Protocol Layer	85
7.1	Parameters utilized in the simulations	95

Introduction

Contents

1.1 Research Objectives	3
1.1.1 Research questions	3
1.2 Approach	3
1.3 Outline	4

Given the increasing number of vehicles on roads worldwide, traffic safety and efficiency are notably crucial aspects in people's daily lives. In particular, traffic jams have been negatively affecting traffic efficiency and leading to a deficit of millions of dollars in many countries, such as The Netherlands, where such problem is severe [TLN 2007]. Air pollution, waste of fuel during the traversal of a congested area and long delays for drivers to arrive, for instance, at work are examples of consequences of having heavy traffic in large cities.

Aiming at addressing this issue, Driver Support Systems (DSS) such as Navigation System and Adaptive Cruise Control (ACC) have been proposed to assist drivers on the road by improving safety, efficiency and comfort in the driving experience. One important benefit of deploying such systems is that by aiding drivers in traversing traffic congestion on highways and consequently having vehicles moving more smoothly close to a constant speed, the emission of CO_2 in the environment is reduced [Johannson 1999].

In the same context, a novel system referred to as the Congestion Assistant has been proposed in [van Driel 2007] by C.J.G. van Driel. In this work, a user needs survey conducted to investigate the perceived needs for driver assistance indicated high user acceptance, in particular, in receiving assistance with the driving task on congested roads. Based on user preferences, the Congestion Assistant has been designed and later evaluated by means of traffic simulation. The assessment of the system focused on traffic efficiency and safety and demonstrated reduction in congestion for all variants of Congestion Assistant considered. Nevertheless, some of its basic operations require information regarding the upcoming traffic condition on the road and no system to acquire it has been proposed.

In particular, the provisioning of up-to-date information about the traffic ahead, *over the driver's horizon*, has been the focus Navigation Systems in the last few years. Great effort has been put for such systems to provide live traffic information in addition to providing guidance simply based on geographical maps and the current location of the vehicle. In their simple form, systems such as TomTom provide an estimate of the time needed to travel through a certain road segment based on data collected anonymously from built-in GPS devices and advice drivers with the probable best (quickest) route to the desired destination. In fact, based on this collected information a prediction of the traffic behavior for each concerned area can be derived for a

short time interval of 15 minutes up to one year in the future as described in [Flow 2009]. Nevertheless, such prediction is often not enough to provide information about unpredictable traffic behavior, such as car accidents. In order to cope with this issue, enhanced systems as the TomTom GO 740 Live [TomTom 2009] promise live real-time information that includes warnings such as “broken down vehicle” or “right lane closed”. The accuracy of the information received, however, is limited by the update interval employed by these systems, which is in the order of a few minutes [Live 2009]. As a consequence, these systems rather provide a rough estimate of the current situation ahead on the road and are not able to capture near instant information of the upcoming road.

An alternative of providing this vision ahead on the road is by using vehicular ad hoc networks (VANETs). This specific type of communication network provides means for delivering the required information either by the support of infrastructure or by exchanging traffic information among vehicles in a multi-hop fashion. VANETs have been of great interest in the last few years due to its large applicability to road safety, traffic efficiency, and entertainment information [Hartenstein & Laberteaux 2008]. A great advantage when compared with approaches used in Navigation Systems is that VANETs rely on a decentralized vehicle collaboration to exchange information. Navigation Systems, on the other hand, depends on the collecting of data from numerous vehicles to be further processed in a centralized database, which makes it less time efficient and scalable. In fact, vehicles equipped with radio transmitting and participating in VANETs are able to directly exchange information in the order of milliseconds with other vehicles within the transmission range utilized. By means of a multi-hop communication, the information may travel up to kilometers in just a few seconds [Wisitpongphan *et al.* 2007].

Given the promising advantages of relying on vehicular communication and the necessity for the Congestion Assistant to acquire information regarding the upcoming traffic, Martijn van Eenennaam has introduced in [van Eenennaam 2008] an efficient means to provide an *over-the-horizon awareness* of traffic jams ahead on the road to vehicles by means of multi-hop vehicle-to-vehicle communication. The proposed solution is a communication protocol that complies with information requirements derived from the Congestion Assistant. The data collected from vehicles ahead regards the current speed profile of the road and it is represented in a structure referred to as TrafficMap. The TrafficMap is built by means of a distributed system called the TrafficFilter and disseminated periodically in network messages to vehicles by means of a directional broadcast protocol. These periodic messages are often referred to as *beacons* that are meant to convey information about the state of the sending vehicle, i.e., position, direction, speed, etc., and possibly also aggregated data regarding the state of its neighbors. A different type are the *Event driven* messages which are triggered on the detection of a hazard, e.g., hard braking from a car, emergency vehicle driving at high speed, etc. In the referred work, because the communication protocol proposed aims at delivering traffic efficiency information to the Congestion Assistant rather than time critical warning of a hazard, only the former type of message is utilized.

The communication protocol proposed by van Eenennaam, however, has been designed with one-dimensional straight highway scenarios in mind. For instance, the information included in the TrafficMap regards a single speed profile of the road ahead, i.e., a combined average behavior of all lanes together. However, this assumption may not be valid in every multiple-lane scenario. In addition, the protocol also does not specifically address situations where multiple information

flows may coexist on a single road, e.g., different messages coming from different roads in a junction point. These are some of the reasons why extensions and/or adaptations might be necessary to make this solution adequate for more real-world scenarios such as multiple-lane roads, junctions, and roads with multiple (opposite) directions.

1.1 Research Objectives

The focus of this research is on assessing the feasibility of the solution proposed in [van Eenennaam 2008] for more realistic scenarios found in highways and proposing extensions and adaptations, whenever necessary. The referred work will therefore serve as the basic starting point. In addition to single-lane roads, the following real-world scenarios are considered: *multiple-lane roads*, *junctions*, and *roads with multiple (opposite) directions*. From this main goal the following research questions regarding each real-world scenario are derived:

1.1.1 Research questions

- *Single-lane highways*: how to address more realistic scenarios in a manner such that the solution does not compromise the overall performance presented in [van Eenennaam 2008] for simple single-lane roads?
- *Multiple-lane highways*: which information must be included from each lane on the road in order to provide an accurate view of the traffic ahead? How to efficiently coordinate the exchange of information in scenarios with vehicles driving nearby in different lanes?
- *Highways with vehicles moving in different directions*: which are the impacts of having vehicles from multiple directions utilizing a vehicle-to-vehicle communication networking solution? Which information and how to efficiently coordinate its exchange among vehicles in order to provide an accurate view of the traffic?
- *Junctions in highways*: which information must be included from each road linked to a junction in order to provide an accurate view of the traffic ahead? How could the exchange of information be coordinated in such scenarios where traffic information may be originated by vehicles from multiple roads?

1.2 Approach

We divide the development of this work into the following tasks:

1. Research and evaluate current vehicle-to-vehicle communication networking solutions proposed in the literature that address the real-world scenarios studied in this work.
2. Study the requirements of each scenario individually and consequently list which information and actions are necessary for our vehicle-to-vehicle communication networking solution.

3. Evaluate the feasibility of the solution proposed in [van Eenennaam 2008] for the mentioned real-world scenarios and whenever necessary propose extensions and/or adaptations for it.
4. Research the feasibility and benefits of such vehicle-to-vehicle communication solution for each described scenario by means of simulations.

1.3 Outline

The remainder of this work is structured as follows:

- Chapter 2 provides important background information about the IEEE 802.11p protocol standard, which represents part of the communication solution we provide in this work.
- Chapter 3 gives a brief overview of the Congestion Assistant and derives the additional information requirements for the addressed real-world scenarios.
- Chapter 4 presents the overview of the solution proposed in this work: the OTHA (Over-the-horizon awareness) protocol.
- Chapter 5 details the upper Traffic Filter Protocol Layer of the OTHA protocol.
- Chapter 6 details the lower Dissemination Protocol Layer of the OTHA protocol.
- Chapter 7 describes and presents the results of the evaluation of our solution performed by means of simulation.
- Chapter 8 concludes this work by answering the research questions raised in this chapter and proposing extensions and improvements as future work.

Background Information: the IEEE 802.11p standard

Contents

2.1 A brief overview on IEEE 802.11	5
2.1.1 Challenges	7
2.2 802.11 in vehicular communication	8

This chapter presents a brief overview of the IEEE 802.11p protocol standard. The standard has been designed to cope with specific characteristics found in vehicular communication and represents part of the communication solution we provide later on in this work. The background information provided here is narrowed down to specific aspects of the standard that are important to the understanding of this work.

The chapter is organized as follows: we start in Section 2.1 by explaining common characteristics of protocols belonging to the IEEE 802.11 family, mostly citing [Schiller 2003, Gast & Loukides 2002]. Later in Section 2.2, the main changes and features added to latest 802.11p drafts with respect to other 802.11 versions are explained.

2.1 A brief overview on IEEE 802.11

IEEE 802.11 is the member of the IEEE 802 family of standards meant for Wireless Local Area Network (WLAN). As part of the IEEE 802 specifications, it focuses on the two lowest layers of the OSI model, i.e., the Media Access Control (MAC) and Physical (PHY) layers. Numerous versions have been designed up to the present moment either to provide improvements for previous versions or tailored features to specific environments. A few examples are versions 802.11b/g/n for general wireless local area networks (WLANs), and 802.11p for vehicular networks. The differences between each variant concern mainly the use of different modulation technique in the physical layer, e.g., Orthogonal Frequency Division Multiplexing (OFDM), Direct Sequence Spread Spectrum (DSSS), or Frequency Hopping Spread Spectrum (FHSS) and the spectrum utilized. Even though there are differences between some versions also in the MAC layer utilized, they all share a basic principle of functioning. In our overview of the 802.11 standard we abstract from the technology employed in the PHY layer and concentrate on basic functions of the MAC layer as it contains more relevant characteristics to the understanding of the solution proposed in this work.

In a 802.11 network nodes must join a Basic Service Set (BSS) which is controlled by a base station called Access Point (AP). The AP is responsible for coordinating the communication among different nodes. In order to extend the coverage of a single BSS, multiple BSSs can establish links between each other by means of an Extended Service Set (ESS). Another type of BSS is the Independent BSS (IBSS) used to establish Ad-hoc networks. Nodes in an IBSS communicate directly with each other and thus must be within direct communication range.

The access to the wireless medium is controlled by one of the following coordination functions: the distributed coordination function (DCF), or the point coordination function (PCF). The former relies on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) access mechanism and may be used either in infrastructure networks with an AP (BSS) or in Ad-hoc networks (IBSS). The latter offers prioritization to different services, however, it requires a central coordination by an AP to determine when nodes are allowed to utilize the wireless medium. Because the solution we propose relies on ad hoc networks, we limit our overview to the CSMA/CA access mechanism.

The CSMA/CA access mechanism has the same principle as the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) employed by the Ethernet protocol (IEEE 802.3 standard) in wired Local Area Networks (LAN). Both rely on the CSMA protocol that works as follows: a node wishing to transmit first sense the medium. If the medium is occupied by other node's transmission, then the node defers its transmission to some time later. Otherwise, the node is allowed to transmit. Collisions occur when multiple nodes sense the medium free and try to transmit at nearly the same time. The difference between both mechanisms lies in ability each one has to deal with collisions. In the case of a wired Ethernet, nodes which are transmitting are still able to listen for incoming signals (collisions) and can send a jamming signal to notify all other stations if a collision has been detected. 802.11 protocols, on the other hand, are half-duplex by design and thus only receivers would be able to detect such collision. However, such detection is not always possible due to the fact that the strongest signal (or the closest source), always dominates the receiver circuitry. Thus, a receiver close to the sender would not be able to receive any other concurrent transmissions thereby being unable to detect collisions [Rayanchu *et al.* 2008]. The 802.11 protocol implements CSMA with Collision Avoidance instead. The receipt of a data packet is confirmed by means of an explicit acknowledgment (ACK) from the receiver. The lack of ACK for a certain transmission gives an indirect indication of a collision.

The functioning of the CSMA/CA access mechanism is depicted in Figure 2.1. A node is allowed to transmit a new packet whenever it senses the medium idle for at least a Distribute Inter-Frame Space (DIFS) period. In the remaining cases that include accessing the medium immediately after a successful transmission, after a retransmission, or when the medium is found busy, the **exponential back-off algorithm** must be executed. In this procedure, nodes will choose a random back-off time from a contention window. This time is defined by a random number of slot times, which have their duration pre-defined in each 802.11 version. Before effectively starting their back-off timer, an additional DIFS period must be waited. After this period, the number of time slots to be waited are decremented only during periods when the medium is found idle. The result is that the back-off timer is stopped in periods when the medium is found busy and resumed when it is idle once again for at least a DIFS period. This mechanism aims at providing some fairness among multiple nodes waiting to access the medium,

since early nodes trying to get their turn to utilize the medium are likely to have their timer expired before others. When the back-off timer is finally over and the medium is idle, nodes may start transmitting. Otherwise, the back-off algorithm is restarted up to a pre-defined maximum number of retries.

The utilization of ACKs in 802.11 protocols function as follows. After each successful reception, the receiver sends an ACK after the Short Inter-Frame Space (SIFS) period as a confirmation to the sender. Since such period is smaller than the DIFS period, receivers always have higher priority when sending ACKs over nodes trying to transmit new packets. When ACKs are not received by senders, it is an indication that some error must have occurred and another attempt is made by retransmitting the last packet. After each retransmission the contention window size from which the node chooses a random number of time slots to wait is exponentially increased in order to reduce the likelihood of another error or collision, thus the name *exponential* back-off algorithm.

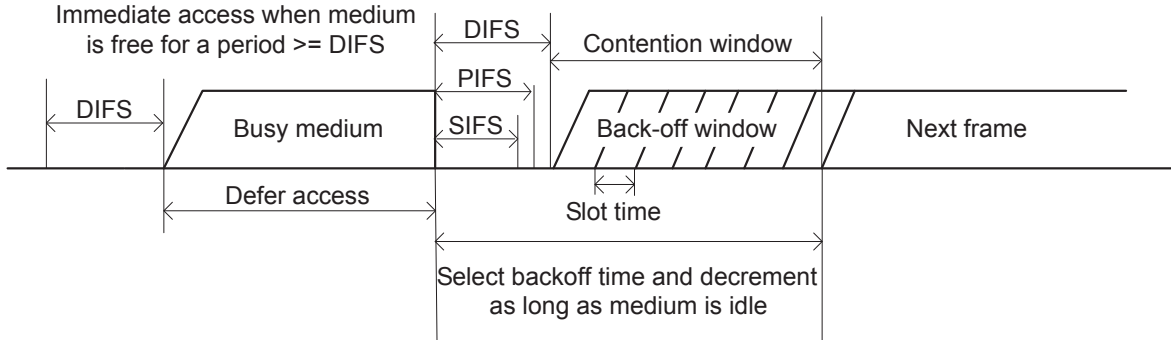


Figure 2.1: IEEE 802.11's Basic Access Method [Schiller 2003]

2.1.1 Challenges

As we will see in following chapters, the communication solution we propose relies on the broadcasting of information to vehicles on the road. Together with its advantages and importance when providing traffic information that may concern all vehicles on the road, the employment of broadcast messages brings several challenges with regard to the reliability of message propagation. When using broadcast, ACKs are not utilized by 802.11 protocols. The reason is that upon the successful receipt of a broadcast message, multiple nodes would reply with an ACK, what increases greatly the number of collisions. The direct consequence of this absence is that nodes are not able to know whether their transmissions have been successfully received by other nodes. To overcome this problem, protocols relying on the broadcasting of messages often infer that a message has been successfully received by other nodes when the transmitting node receives an echo of that message from other nodes, in case the rebroadcast of messages is defined by the protocol. Nevertheless, it is still uncertain whether or not all neighbors have received the message.

Another major existing problem in wireless environments is the *hidden terminal problem* as illustrated in Figure 2.2. In this scenario, B is within the transmission range of A while C is placed outside, i.e., C cannot sense ongoing transmissions from A. From the perspective of node A, node C is a “hidden” node. The hidden terminal problem occurs when C senses the medium idle and start transmitting a message to node B or any other node. This additional transmission causes a collision in B due to its presence in both transmission ranges of A and C.

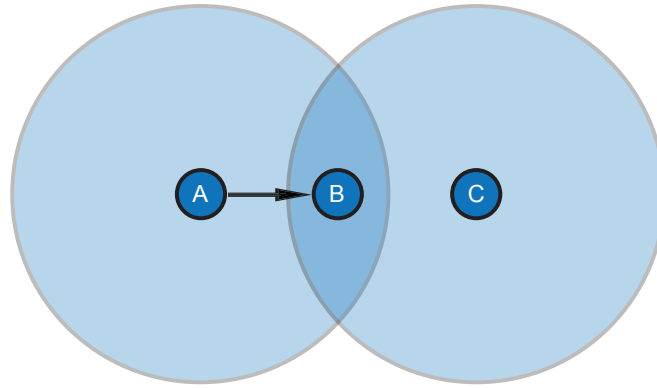


Figure 2.2: The Hidden Terminal Problem

An extension referred to as the Request to Send (RTS) and Clear to Send (CTS) mechanism has been introduced to 802.11 protocols. The RTS and CTS signals are used to inform other nodes about requests and the existence of transmissions. Due to their small size and thus quick transmissions, the RTS/CTS mechanism reduces the probability of collisions caused by the hidden terminal problem. However, it introduces an additional delay before each transmission and collisions may still occur among RTS/CTS messages. Despite this its advantages and disadvantages, this mechanism is not available for broadcast messages. As a consequence, the hidden terminal becomes one major problem when relying on the broadcasting of messages.

One final challenge regards the Contention Window size. As explained previously, when ACKs are not received by senders some error is assumed to have occurred and another attempt is made by retransmitting the last packet. After each retransmission, the contention window size is exponentially increased in order to decrease the probability of collisions. However, because of the lack of acknowledgments when performing a broadcast, the Contention Window is actually never increased. This constant Contention Window size may severely harm the reliability of the communication, as in certain versions of the protocol the minimum size set is rather small, e.g., limited to 16 time slots.

2.2 802.11 in vehicular communication

Due to specific challenges vehicular environments impose on current wireless communication systems, there has been efforts over the last years in establishing a new and adequate communication standard exclusively meant to address vehicular communication. One example of such effort is the convergence of the US Department of Transportation and the E. U. CAR 2

CAR Communication Consortium (C2C-CC) to use the IEEE 802.11p WAVE standardization [IEEE 2006]. The IEEE 802.11p WAVE standardization process originates from the allocation of the Dedicated Short Range Communications (DSRC) spectrum band in the United States and the effort to define the technology for usage in the DSRC band. The standard defines data rate from 3 to 27 Mbps in networks with high mobile nodes, moving at speeds greater than 60 mph.

In the US, the Federal Communication Commission allocated 75 MHz of Dedicated Short Range Communications (DSRC) spectrum at 5.9 GHz to be used exclusively for vehicle-to-vehicle and infrastructure-to-vehicle communications [Jiang & Delgrossi 2008]. As shown in Figure 2.3, the DSRC spectrum is structured into seven 10 MHz wide channels. Channel 178 is the control channel (CCH), which is restricted to safety communications only. The remaining channels are service channels (SCH) available for both safety and non-safety usage, with the exception of channels 172 and 184 which are reserved for special purposes.

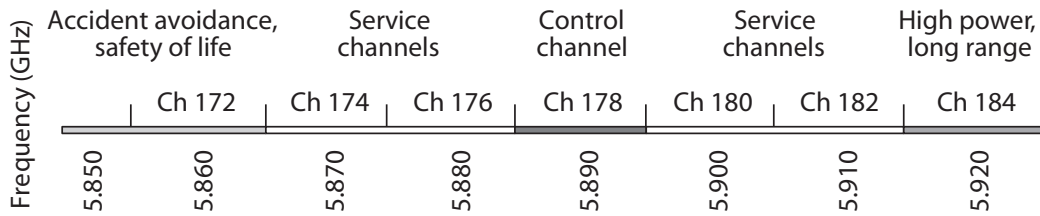


Figure 2.3: DSRC spectrum band and channels in the US

The IEEE 802.11p standard follows the scope established for every IEEE 802.11 standard and, therefore, addresses strictly Media Access Control (MAC) and Physical (PHY) layers. The necessity of a new and specific protocol specification for vehicular networks lies in the fact that current 801.11 specifications for MAC and PHY layers have not been designed to cope with the constant dynamic movement of vehicles. IEEE 802.11p is essentially based on IEEE 802.11a with adjustments for low overhead operations in the DSRC spectrum.

The following adjustments have been proposed in order for the 802.11p MAC and PHY layers to be suitable for vehicular networks:

- **MAC Layer:** changes focus on providing a very efficient communication group setup without much of the overhead typically needed in current IEEE 802.11 MAC standards. Current 802.11 MAC protocols rely on an Infrastructural Basic Service Set (BSS) which is a group of nodes accessing a common Access Point (AP). The BSS mechanism controls access to an AP's resources and services, and also allows for a radio to filter out the transmissions from other unrelated radios nearby. When nodes want to join a BSS, they must first listen for beacons from an AP and then join the BSS by means of numerous interactive steps including authentication and association. A similar mechanism referred to as Independent BSS (IBSS) is offered to Ad Hoc networks. Due to the high latency and overhead when establishing connections to a BSS, this mechanism is not suitable for vehicular networks. The following changes are proposed to mitigate these problems, as detailed in [Jiang & Delgrossi 2008]:

- *WAVE mode*: a node in WAVE mode can transmit and receive data frames without the need to join a BSS. In this way, vehicles can immediately communicate with each other upon encounter without any additional overhead as long as they operate in the same channel using a special BSS identification (BSSID) referred to as wildcard BSSID.
 - *WAVE BSS*: this new type of BSS referred to as WBSS allows vehicles to join a WBSS by only receiving a WAVE advertisement with no further interactions. Such advertisements are used on demand by upper layers above the IEEE 802.11 and contain all information necessary for vehicles to join a WBSS.
 - A node currently belonging to a WBSS is still in WAVE mode and, therefore, can still transmit frames with the wildcard BSSID in order to reach neighboring nodes in cases of safety emergencies.
- **PHY Layer**: while changes in the MAC layer are translated in new software, any change in the PHY layer requires updates in existing hardware and should be avoided whenever it is possible. The changes proposed are meant to provide vehicles with an effective communication among fast moving vehicles in the road. To achieve such goal, the following changes are proposed:
 - IEEE 802.11p is essentially based on the OFDM PHY defined for IEEE 802.11a, with a 10 MHz wide channel instead of the 20 MHz. The reasoning behind this option is that the guard interval defined at 20 MHz is not long enough to prevent inter-symbol interferences within the vehicle's own transmissions in vehicular environments. Since 802.11 already defines a 10 MHz channel, the implementation of this new setting in 802.11 hardwares is straight-forward.
 - Because of the high proximity of vehicles on the road, IEEE 802.11p introduces, even though outside of its scope, some improved receiver performance requirements in adjacent channel rejections.

The Congestion Assistant Requirements

Contents

3.1	System Overview	11
3.2	Requirements for Real-World Scenarios	14
3.2.1	Multi-lane highways	14
3.2.2	Roads with different directions	15
3.2.3	Junctions	17
3.3	System Requirements	19
3.4	The Navigation System	21
3.4.1	Positioning sensor	21
3.4.2	Speed sensor	22
3.4.3	Headway sensor	22
3.4.4	Lane sensor	22
3.4.5	Geographical map	23

This chapter presents a brief overview of the Congestion Assistant and derives the additional information requirements for the addressed real-world scenarios.

The chapter is organized as follows: Section 3.1 introduces the basic structure of a vehicle equipped with Intelligent transportation systems (ITS), which includes the Congestion Assistant. In addition, it provides an overview of the information requirements for the Congestion Assistant outlined in [van Eenennaam 2008] for single-lane road scenarios. In the sequel, Section 3.2 motivates and introduces additional requirements for the following real-world scenarios: multi-lane highways, roads with different directions, and junctions. Section 3.3 provides the overall system requirements. Finally, Section 3.4 gives examples of how the Navigation System can provide part of the information required by the Congestion Assistant.

3.1 System Overview

The Congestion Assistant is a system proposed in [van Driel 2007] by C.J.G. van Driel which aims at supporting drivers in traffic congestion situations on highways. It is one of the many *Intelligent Transportation Systems (ITS)* [IEEE 2009] that are expected to equip vehicles in order

to improve traffic safety and efficiency in the next couple of years. Intelligent transportation systems (ITS) comprise a broad range of wireless and wire line communications-based information and electronics technologies, such as Driver Support Systems (DSS).

A simplified abstraction of vehicles equipped with such intelligent systems can be depicted as follows in Figure 3.1. Each vehicle contains two main boxes that represent the separation of the *Application* and *Communication* layers. The application layer includes the mentioned Intelligent Transportation Systems such as the navigation system, the **congestion assistant**, and the collision avoidance system of a vehicle. These systems have required information to be obtained either by means of internal sensors/receivers or external sources. The internal sensors/receivers are part of the application layer and are usually components of the intelligent transportation systems. Examples of such sensors/receivers are the GPS receiver and sensors for collision avoidance and lane detection. The communication layer is then responsible for providing the required information regarding external sources, which in this case are other vehicles and/or infrastructure on the highway. Communication protocols designed to control and filter the exchanging of messages among vehicles are examples of components that might be part of the communication layer.

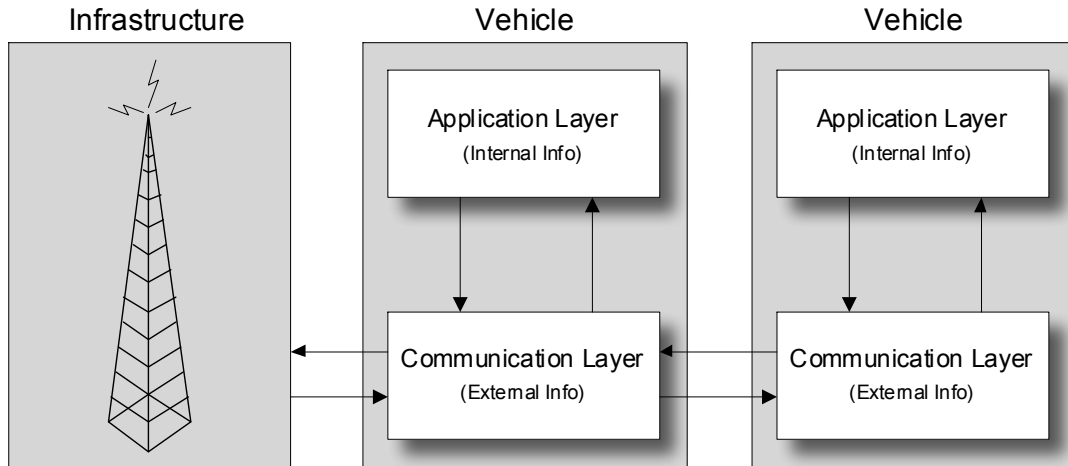


Figure 3.1: Abstract of Intelligent Transport Systems within a vehicle

Accordingly, the Congestion Assistant has required information that may be obtained both from the sensors/receivers in the *application layer* and from the components of the *communication layer*. Van Eenennaam in [van Eenennaam 2008] analyzes the system and outlines the required information considering a one-dimensional straight highway, which we summarize in the following.

The Congestion Assistant system performs three tasks: Warning & Information (W&I), Active Pedal (AP), and Stop & Go (S&G), as depicted in Figure 3.2. The W&I informs the driver about the traffic conditions ahead. Upon the receipt of a message regarding the upcoming traffic, the driver is able to prepare for congestion and will receive update information during his traversal through the current traffic jam, or may even choose an alternative route. The following information has been outlined for the basic functioning of W&I:

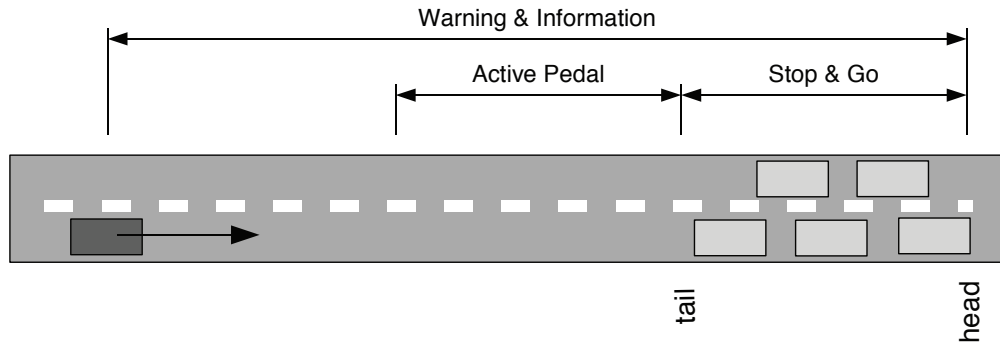


Figure 3.2: Overview of the Congestion Assistant functioning [van Eenennaam 2008]

- Own position, speed
- The position of the tail and head of the jam
- Average speed of the jam, movement within the jam

With the own driver's position and the position of the head and tail of the jam, the total length of the jam and also the distance before the driver will reach the congestion are derived. The average speed can be used to estimate the expected additional delay due to the traffic congestion.

The second task, the AP, gives a counter pressure on the accelerator pedal in order to induce the driver to gradually reduce the vehicle's speed starting from a safe distance before reaching the tail of the traffic jam. This can prevent vehicles from braking dangerously and at the same time keep a smooth inflow of traffic, which is desirable to avoid a rapid growth of the jam. The AP needs the information:

- Position and speed of the vehicle
- Position and speed of the tail of the jam

Based on this information, the AP can safely calculate the distance and time before the congestion that it has to be activated in order to reduce the vehicle's speed up to the speed of the tail of the jam.

The remaining task of the Congestion Assistant is the S&G function. As soon as the vehicle enters in the traffic jam, S&G will assist the driver by automatically accelerating and reducing speed more smoothly when compared with how human drivers would normally behave. After the vehicle traverses the traffic jam, the S&G is disengaged and manual driving recommences. The following data is required:

- Position of the head and tail of the jam
- Speed to be maintained during the traversal of the jam

- Distance to vehicle in front

The position of the tail and head of the jam serve to know when the S&G must be engaged and disengaged. The speed to be maintained must be based on traffic flow research, since it may vary depending on the current road condition [van Driel 2007]. Finally, the distance to the vehicle in front is necessary to keep vehicles separated by a safe margin distance.

The outlined required information are classified into either the application or communication layers as explained previously. Therefore, required information obtained locally by means of the explained internal sensors/receivers in the vehicle are classified as being generated by the *application layer* whereas required information obtained by external means, e.g., other vehicles, are classified as being generated by the *communication layer*. This is shown in Table 3.1.

<i>Application layer</i>	<i>Communication layer</i>
Own speed	Speed of tail of the jam
Own position	Speed of head of the jam
Distance to vehicle in front	Distance to vehicle in front
	Position of the tail of the jam
	Position of the head of the jam

Table 3.1: Classification of the required information

3.2 Requirements for Real-World Scenarios

When extended to more realistic situations, the required information described in Section 3.1 may have to be updated to consider different types of road environments. One important aspect to evaluate is whether regarding the road as a pipe with a certain flow speed as proposed in [van Eenennaam 2008] by Van Eenennaam is still a valid approach in the new scenarios considered in this work. Therefore, in every task defined by the Congestion Assistant, additional information may be necessary for drivers to receive an overview of the upcoming traffic condition in order to safely take actions before in fact reaching traffic jams. This information must be as accurate as possible for each new situation.

In this work, the following scenarios are considered: *multiple-lane highways*, *roads with different directions*, and *junctions*. These are only a fraction of possibilities that are found in highway scenarios. We restrict ourselves to the mentioned scenarios and motivate as well as outline the additional information required for each of them. Therefore, in the following subsections all required information outlined so far are maintained and our focus will be on the additional information required for each mentioned scenario.

3.2.1 Multi-lane highways

Generally, highways are the main roads used to connect important destinations, such as cities and towns. Highway designs may vary considerably depending on each country. The number of lanes contained in each highway can also vary from two up to multi-lane freeways. In fact,

the widest highway (maximum number of lanes) is The Katy Freeway (part of Interstate 10) in Houston, Texas, United States of America, with a total of 26 lanes in some sections when considering both directions and counting auxiliary lanes [KatyFreeway 2009].

In a multi-lane highway, the current condition, e.g., flow speed, of each lane may change over-time, which makes it difficult to define a common behavior for the whole road. Although a maximum speed value is defined in each highway segment for all the present lanes, there is a noticeable variation on the minimum speed average depending on the lane position. In most countries, the right-hand traffic regulation dictates the minimum speed average to increase from the rightmost to the leftmost lane positions. Highways in the remaining countries, which include England and Japan, have left-hand traffic regulations and thus the opposite pattern with averages being increased from the leftmost to the rightmost lane positions. Right-hand traffic and left-hand traffic regulations define that all traffic must be kept either on the left or the right hand side of the road. For instance, in the United Kingdom the *Highway Code* [Motorists 2007] states the following:

136. Once moving you should keep to the left, unless road signs or markings indicate otherwise. The exceptions are when you want to overtake, turn right or pass parked vehicles or pedestrians in the road.

Therefore, it is reasonable to predict that the *speed profile* of the present lanes in a highway will differ depending on their positions.

In addition to the mentioned enforced speed restrictions of lanes in a highway, there exist situations where the difference of speed profile between lanes becomes more evident. Consider the scenario illustrated in Figure 3.3(a). In this example, the first lanes on the left L_0 and L_1 are congested whereas the remaining lanes L_2 and L_3 have a free-flow, i.e., non-congested, access to a new road direction. This road splitting is a very common situation found in city entrances where the highway is divided into roads with different directions in order to distribute the traffic to various parts of the city where drivers may want to go (Figure 3.3(b)).

In order to provide vehicles with an extended view up to a few kilometers ahead on the road, vehicles must obtain information regarding each lane individually. A speed profile summary of the entire highway regarding the road as a pipe is therefore not enough to depict the current traffic condition.

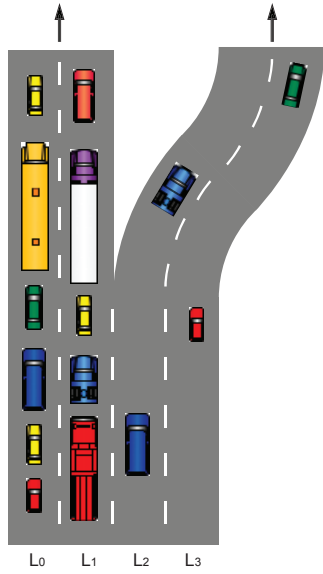
The following additional information is then required to provide such awareness to vehicles:

- Own lane number
- Lane number(s) of the traffic jam

By having the current lane number of vehicles and the lane number(s) on which the traffic jam is located, the Congestion Assistant is able to warn drivers about upcoming traffic jams so they can decide whether or not to move to a different lane accordingly.

3.2.2 Roads with different directions

In addition to multi-lanes, another basic scenario considered in this work is the existence of roads with different directions. In such scenarios, the Congestion Assistant must be able to uniquely



(a) Congested and free-flow lanes



(b) City entrance in São Paulo, Brazil [Ortiz 2008]

Figure 3.3: An illustration and a real example of multi-lane highways

identify each lane of the roads in the current *road segment*. We refer to the road segment as the region for which the Congestion Assistant will provide awareness to drivers. This could comprise more than one road and multiple directions and could be delimited, for instance, by the road length, area of interest, or time interval. The reason why we limit the awareness provided up to a certain delimitation (road segment) lies in the fact that the information regarding traffic, e.g., in hundreds of kilometers, ahead on the road might be outdated upon the receipt of messages regarding such information. Another reason is that such information may not even be interesting to drivers anymore as they may move to a completely different direction.

More specifically, the Congestion Assistant must recognize the exact location of a traffic jam. In a situation such as the one depicted in Figure 3.4, vehicles might obtain information regarding various directions and not only from the common positive (current) and negative (opposite) directions of a highway. The system must be able to base its internal decisions on the *origin* of the information.

In order to identify the source location of the information, the following information is required:

- Road identification of the traffic jam
- Road segment
- Own Road Identification
- Own driving direction
- Own lane number



Figure 3.4: Example of a spaghetti junction [Cozart 2000]

- Lane number(s) of the traffic jam
- Driving direction of the traffic jam

With the road segment, the road identification, direction, and the lane numbers on which the traffic jam occurs, the Congestion Assistant is able to uniquely identify the location of current congestions within the road segment. Moreover, based on the own vehicle's direction, road identification, and lane number, proper actions may be taken to warn and advise drivers for occasional traffic jams on the current highway.

3.2.3 Junctions

As described previously in Section 3.2.2, due to the large number of roads with different directions that can exist in highways, an accurate identification of the location of congestions in the current road segment is required by the Congestion Assistant in order to take proper actions when warning drivers. Nevertheless, we can reduce the road segment by including only information regarding the location of traffic jams that drivers are actually able to go. In fact, what dictates the possibilities of direction drivers have ahead on the highway are the *junctions*.

Road junctions are locations of convergence and divergence of multiple roads. There are two main different types of junction between roads: *interchanges* and *intersections* [Wikipedia 2007]. The former comprises junctions where roads pass above or below one another, preventing a single point of conflict by utilizing grade separation and slip roads. These are the typical junctions found in highways. Oppositely, junctions of the latter type do not use grade separation (they

are at-grade) and roads cross directly. Intersections are commonly used within cities. Since our focus is on highways, only interchanges will be considered.

In this scenario, the Congestion Assistant needs to know the location of junctions on the road and what is the traffic condition in each possibility of direction ahead. Therefore, the system requires the information about the location of junction points where the highways are diverged or merged. There are two kinds of points: the *exit* and *entrance* points. The system needs to obtain only the location of exit points, as entrance points concern roads joining the current driver's highway and hence are not possibilities of direction for vehicles to go. For the sake of clarification, Figure 3.5 illustrates the exit and entrance points with respect to road R_2 . From the point of view of vehicle C_1 in road R_2 , the dashed area is an exit point area and a possibility of moving to road R_1 . Thus, only traffic information regarding that point onwards is required. Vehicle C_1 , however, does not require the information regarding the entrance point area of vehicles coming from road R_1 to R_2 , as clearly this is not an option of direction to go.

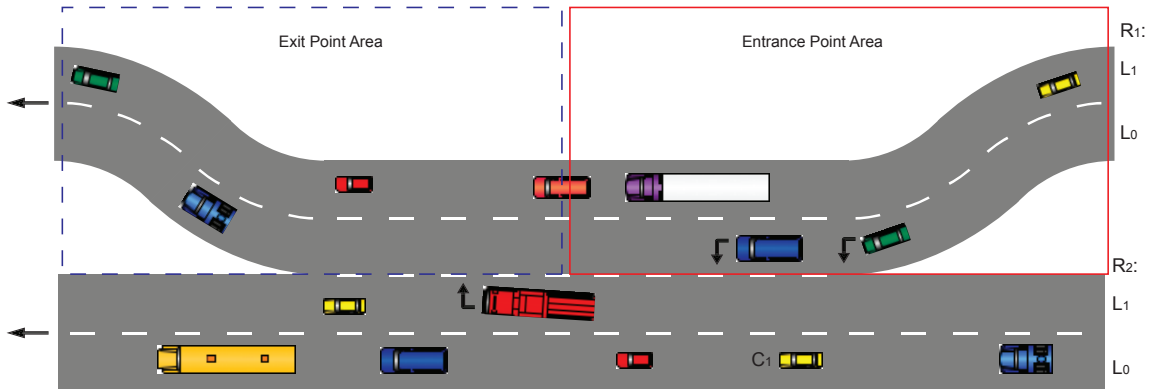


Figure 3.5: Example of a junction with entrance and exit points

An abstraction of junctions in highways is depicted in Figure 3.6. In this example, vehicle C_1 has several possibilities of directions to take along the highway R_1 . The white boxes where different roads meet represent the junctions in this road segment.

The required information updated to include junction scenarios as described in this section are listed as follows:

- Junction point location
- Type of junction point: *exit* or *entrance* point
- Road identification of the traffic jam
- Road segment
- Own Road Identification
- Own driving direction
- Own lane number

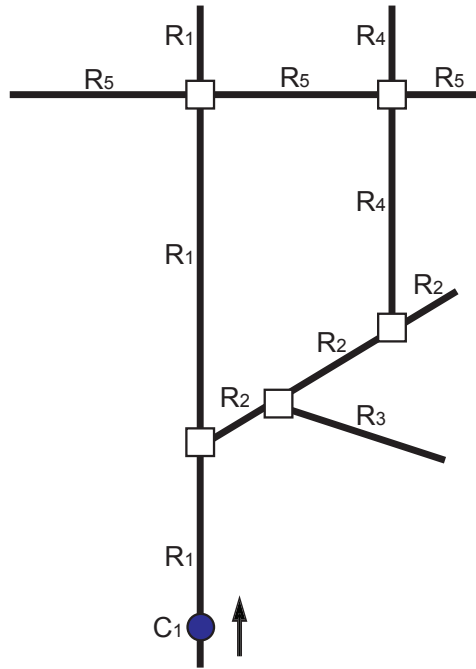


Figure 3.6: Abstraction of junctions in highways

- Lane number(s) of the traffic jam
- Driving direction of the traffic jam

In this list, the required information explained in the previous section is maintained in order to identify the location of traffic jams uniquely in a road segment. The junction point location is now required for the Congestion Assistant to know where junctions are located and which are the possibilities ahead on the highway. Since only exit points are of interest, the type of junction is also required.

3.3 System Requirements

After motivating and outlining the additional information required for each new scenario considered, we classify them either belonging to the *application* or *communication* layers. Table 3.2 lists the required information already outlined in Table 3.1 (page 14) by van Eenennaam in [van Eenennaam 2008] combined with the new requirements described in the previous few sections. Notice that accordingly to the description of each layer described in Section 3.1, required information obtained internally is classified into the *application* layer while required information obtained externally are classified into the *communication* layer.

The Congestion Assistant is envisioned to be structured as shown in Figure 3.7. We assume in this work that the system will interact with a Navigation System on-board in the application layer. Within the Navigation System, a number of components, namely, a *Positioning Sensor*,

<i>Application layer</i>	<i>Communication layer</i>
Own speed	Speed of tail of the jam
Own position	Speed of head of the jam
Distance to vehicle in front	Distance to vehicle in front
Own lane number	Position of the tail of the jam
Road segment	Position of the head of the jam
Own Road Identification	Lane number(s) of the traffic jam
Own driving direction	Road identification of the traffic jam
Junction point location	Driving direction of the traffic jam
Type of junction point	

Table 3.2: Required information considering the new scenarios described

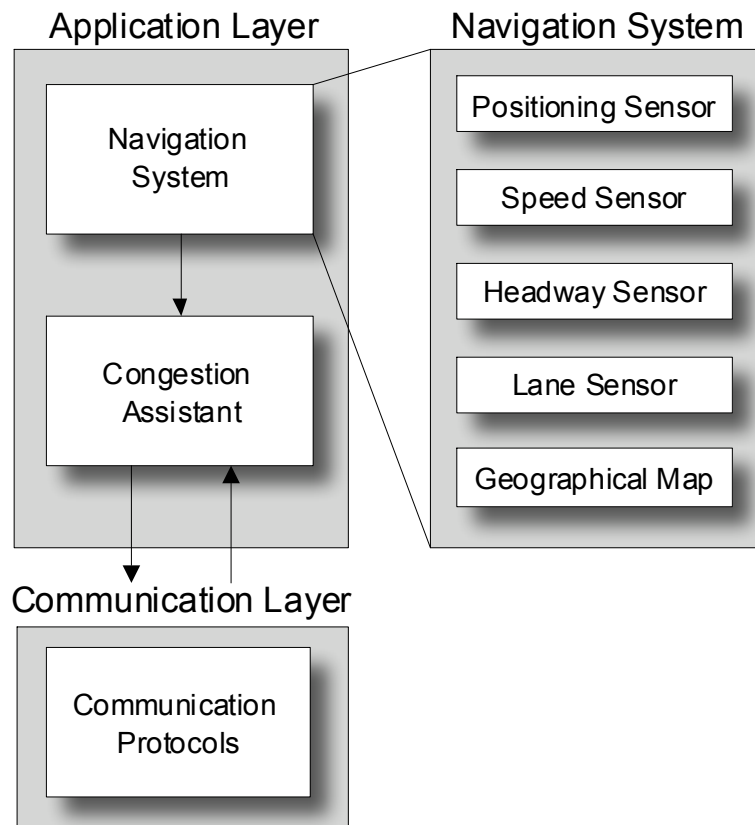


Figure 3.7: Interactions between the Congestion System and its required applications

a *Speed Sensor*, a *Headway Sensor*, a *Lane Sensor*, and a *Geographical Map* would provide the following information:

- **Positioning sensor:** the geographical position of the vehicle.
- **Speed Sensor:** the current speed of the vehicle.
- **Headway Sensor:** the distance to the immediate vehicle in front on the same lane on the road.
- **Lane Sensor:** the current lane number of the vehicle.
- **Geographical map:** road identification (ID), road direction, location of junction points and junction point type.

3.4 The Navigation System

Automotive navigation systems [Ayanoglu & Sabnani 1997] have become popular in assisting drivers on the road by providing directions to specific destination locations defined by the driver. They are generally equipped with a GPS receiver that acquires up-to-date position data which is used to locate the vehicle in the geographical map stored in a local database, among other features. We envision that the Congestion Assistant will interact and acquire required information directly from a navigation system on-board the vehicle. In the remainder of this section we provide examples of how the Navigation System can provide part of the required information as described in Section 3.3 by means of the following components: *positioning sensor*, *speed sensor*, *headway sensor*, *lane sensor*, and *geographical map*. Rather than presenting a comparison between different methods of obtaining such required information, our objective here is limited by simply motivating their existence.

3.4.1 Positioning sensor

Due to its wide availability and current low cost, the Global Positioning System (GPS) [Hofmann-Wellenhof *et al.* 2001] sufficiently provides the location of vehicles on the road required by the Congestion Assistant. The system which has priorly been employed in military applications became popular in most automotive navigation systems and recently in mobile phones as well.

In order for vehicles to receive up-to-date information about their location, a GPS receiver equipping them calculates the current geographical position by precisely timing the signals sent by the GPS satellites high above the Earth. These signals which are constantly sent contain information about the time they were sent, precise orbital information, and the general system health and rough orbits of all GPS satellites. The GPS receiver then measures the transit time of each message and computes the distance to each satellite. Finally, the location is derived by means of geometric trilateration that combines these distances with the location of the satellites. The accuracy of these measurements may vary from a few meters to tens of meters [Taylor *et al.* 2006].

3.4.2 Speed sensor

The current speed of vehicles is generally provided by a speedometer [Webster 1978]. A speedometer is a device that measures the instantaneous speed of a land vehicle. In short, speedometers measure the rate of rotation of a wheel or fan whose rate of rotation depends on the speed of the vehicle. Most speedometers have accuracy error tolerances of some 10% plus or minus due to wear on tires as it occurs, although it is argued in [Victoria 1994] that a combination of factors including for instance, the changing of wheel or tire sizes, can lead to errors of 15 km/h or more.

3.4.3 Headway sensor

The distance to vehicles in front can be provided by the use of *forward-looking radars* [Witte 1992]. An example of such radar is described in [Farkas *et al.* 1997]. By means of echoes, the system senses vehicles and other obstacles in the lane ahead on the road. This mechanism is generally envisioned to be included into Adaptive Cruise Control system in order for vehicles to automatically adjust their speed and maintain a proper distance from other vehicles on the same lane ahead [Marsden *et al.* 2001].

3.4.4 Lane sensor

Lane detection has been subject of study in the last few years given the inaccuracy present by current GPS systems. The identification of lane has become important in complementing other sensors, such as the GPS, in accurately defining the location of vehicles on the road. Generally, the methods proposed in the literature rely on image processing algorithms in order to provide such identification. The ultimate goal is to identify the current lane quickly regardless of adverse conditions that may present on the road, i.e., different brightness or shadows, painted or unpainted roads, curve or straight roads.

The work presented in [Chausse *et al.* 2005], for instance, proposes a combination of GPS absolute localization with data computed by a vision system giving the position and orientation of the vehicle on the road. The overall precision provided by this method has shown to perform better than the GPS alone: a low cost GPS has a 20 meters of precision while their estimates have at best 48 centimeters of precision along the road axis and 8 centimeters of precision perpendicularly to the road axis.

The method proposed in [Danescu *et al.* 2007] combines stereovision-specific techniques with grayscale image processing for maximizing the robustness and applicability against the difficult conditions of the urban environment. Other approaches in [Zhang *et al.* 2005, Kim *et al.* 2007] rely on support vector machine (SVM) to recognize lane color robustly for various lighting conditions including shadow, backlight, sunset, and so on. In particular, the method presented in [Kim *et al.* 2007] combines the information obtained from the lane sensor utilized with the navigation database in a way it is possible to accurately define whether the vehicle is currently located in the leftmost, middle, or the rightmost lane.

Automakers, such as Toyota in [Toyota 2009], have gradually introduced consumers to products with *Lane-Keeping Assist* technologies. Even though these products aim at helping drivers to keep vehicles stay on course by recognizing the lane with a lane recognition camera rather

than enhancing the vehicle's location, such technology also rely on sensors and identification of the current lane on the road.

3.4.5 Geographical map

As part of most Navigation Systems, geographical maps provide enhanced guidance to drivers with the matching of the current vehicle's coordinates with the corresponding available geographical map segment. From this matching, the Congestion Assistant is able to acquire road IDs, road directions, location of junction points and their type, as motivated as information requirements for its proper functioning. In order to provide an example of how this information can be acquired from such geographical maps, we outline in this section how this information is represented and described in the Geographic Data File (GDF) standard. GDF is a map format standard defined by the International Organization for Standardization (ISO) and it is used to describe and exchange road network-related data [Telematique 2009]. In fact, Major map vendors such as TeleAtlas [Tele-Atlas 2009] and NAVTEQ [NAVTEQ 2009] provide maps in GDF.

According to the draft GDF standard specification in [ISO 2002], the following information specification is provided:

- **Route Number:** The Route Number is the ID number of a particular route in a given road network as attributed by a national, sub-national or international organization (e.g., the numbering of the departmental roads in France or the E-roads in Europe). It is one example of a unique road identification in highways.

Example: E35 and A2 in The Netherlands.

- **Composite Exit Number:** It is the Information about numbers and names of an individual exit along a freeway. It has as sub-attributes the Exit Number, Official Name, Alternate Name and the Route number. It serves as the identification of exit points in highways.

Example: Exit 2 at E35, Exit 34B at A2.

- **Exit at Interchange:** It is the relation between an Interchange and one or more contained Junctions which correspond to an exit specified by means of one Composite Exit Number. For each traveling direction two exit points exist and the two exits have distinguished exit numbers (e.g., Exit 5 North and Exit 5 South). The combination of the Composite Exit Number and Exit at Interchange values provide a complete knowledge of exit points identification and location at junctions in highways. Since exit point is the junction type of interest (see Section 3.2.3), these values suffice for the correct provisioning of the location and type of junction points in highways.
- **Direction of Traffic Flow:** It is the direction(s) of traffic flow allowed on a road, and consequently it meets our requirement for the road direction. The following identification is possible:

- Traffic is allowed in both directions.

- Traffic is closed in the positive direction, and open in negative direction.
- Traffic is closed in the negative direction, and open in positive direction.
- Traffic is closed in both directions.

The Over-the-Horizon Awareness Protocol Overview

Contents

4.1 The Protocol Overview	25
4.2 Communication Requirements and Assumptions	28
4.2.1 Assumptions	28
4.2.2 Communication Requirements	28

Based on the information requirements described in the previous chapter, we propose a solution to provide the Congestion Assistant with the required vision of the upcoming road condition. Our solution is a communication protocol, the Over-The-Horizon Awareness (OTHA) protocol, that altogether comprises extensions and modifications to the communication protocol presented in [van Eenennaam 2008]. This chapter presents an overview of the OTHA protocol by describing its basic characteristics, goals, and interactions with other systems that complement it. Further details will be given in following chapters.

The chapter is organized as follows: Section 4.1 presents an overview of the OTHA protocol with its main goals and characteristics; in the sequel Section 4.2 describes the assumptions and communication requirements of the OTHA protocol.

4.1 The Protocol Overview

The main goal of the OTHA protocol is to provide the Congestion Assistant with information about occasional traffic jams that might be present ahead on the road. One way of accomplishing this task is to deliver information exclusively about the beginning and end of traffic jams. In this work, however, we rather focus on providing a *speed profile* of roads ahead that includes both information about traffic jams and also about the current overview behavior of vehicles in a certain area. This latter approach meets our main goal and yet it is capable of offering an extra view of the speed pattern occurring on each upcoming road. The judgment of what or where there is in fact a traffic jam is left for the Congestion Assistant to make. In fact, the OTHA protocol merely provides traffic information to the Congestion Assistant, meaning that it does not influence on the Congestion Assistant's decisions on how the information will be utilized.

In addition, because the OTHA protocol aims at delivering traffic efficiency information to the Congestion Assistant rather than time critical warning of a hazard, our protocol does not concern time critical applications. For this reason, the protocol must give other more critical

applications the opportunity to use the radio channel. On the other hand, the delay in receiving new traffic information must be sufficiently short so that the information received is still fresh and accurate.

The underlying idea behind the OTHA protocol is depicted in Figure 4.1. By relying only on their human eyes, drivers would only be able to identify the upcoming congested area at the moment they join the traffic jam. By means of the OTHA protocol, the communication among vehicles provides an over-the-horizon awareness, what would allow the Congestion Assistant to accurately assist drivers before and during the traversal of the traffic jam, as explained in Chapter 3.

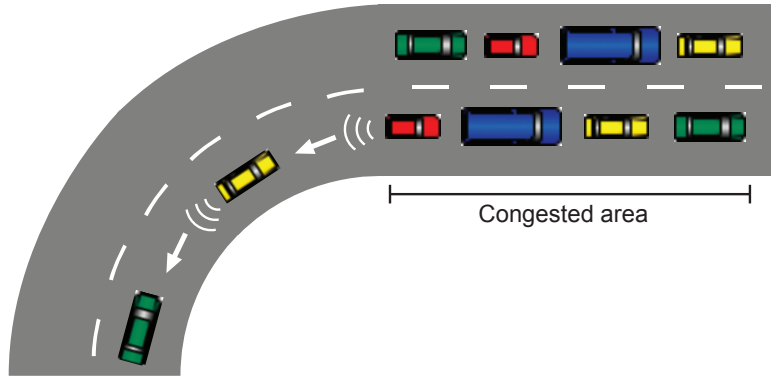


Figure 4.1: Basic functioning of the OTHA protocol

The overview of the OTHA protocol and its interactions with the Congestion Assistant and other components is illustrated in Figure 4.2. The OTHA protocol is placed within the communication layer and provides up-to-date information about the traffic ahead on the road to the Congestion Assistant placed in the upper application layer. In order for the OTHA protocol to provide a speed profile of roads ahead, the internal information outlined in the previous chapter 3 has to be obtained from a Navigation System present in the application layer. The Navigation System also provides information such as a speed, position, direction, and distance to vehicle in front, to the Congestion Assistant as outlined as requirements for its correct functioning. Within the communication layer, the OTHA protocol functions as network and application network protocol layers in the OSI reference model for telecommunications [Wetteroth 2001]. The IEEE 802.11p [IEEE 2006], the upcoming IEEE standard for vehicular communication, complements the communication layer by providing the Link and Physical layers.

The OTHA protocol is divided in two layers: the *Traffic Filter Protocol Layer* and the *Dissemination Protocol Layer*. The goals and characteristics of each layer are described as follows:

- **Traffic Filter Protocol Layer:** the goal is to provide an *accurate* view of the traffic ahead. Information about the upcoming traffic is obtained from the lower Dissemination Protocol Layer and the following main tasks are accomplished:
 - It manages and organizes the traffic information received in a structure referred to as

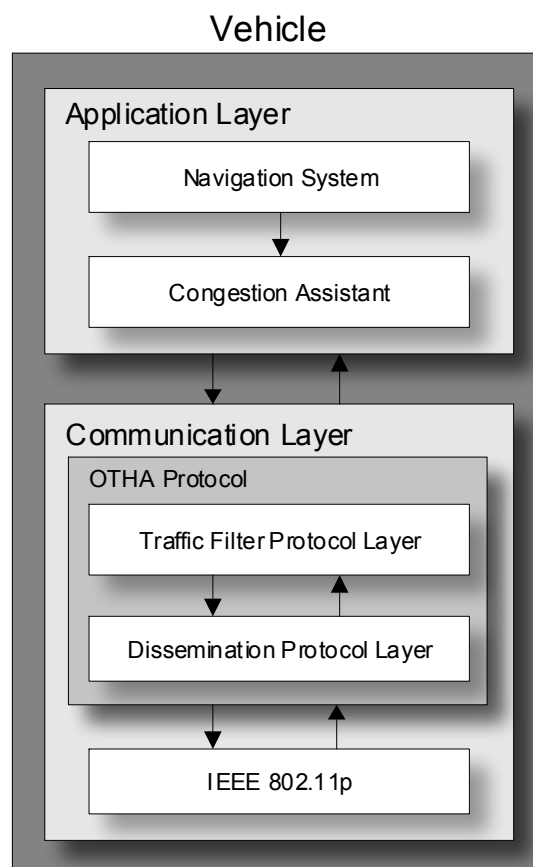


Figure 4.2: Overview of the Over-the-Horizon Awareness (OTHA) Protocol

the *TrafficMap*.

- It filters the traffic information to include only essential characteristics of the road ahead.
 - The distance up to where the awareness is provided is limited and defined by the *road segment*. The *road segment* is an area pre-defined by the Congestion Assistant, e.g., 10 x 10 km².
- **Dissemination Protocol Layer:** the goal is to disseminate the traffic information managed by the upper Traffic Filter Protocol Layer as quick and efficient as possible. This can be translated into the following desired characteristics: *low end-to-end delay* (the information must be fresh), *high reachability* (all target vehicles must receive traffic information), *low channel load introduced* (other applications must be able to also utilize the radio channel). Its main tasks are:
 - It builds messages referred to as *TrafficMap Messages* with the most up-to-date traffic information to be sent to other vehicles.
 - It manages the exchange of TrafficMap Messages among vehicles in an ad-hoc fashion, thus, in a vehicular ad-hoc network (VANET).
 - It relies on a directional broadcasting algorithm to disseminate TrafficMap Messages to every vehicle that it concerns. This is defined by the *message direction*. In the scope of this work, we simply define the message direction as being upstream the road (behind the vehicle).

4.2 Communication Requirements and Assumptions

4.2.1 Assumptions

For the sake of simplification, the OTHA protocol is designed upon the following assumptions:

- Every vehicle is equipped with a wireless radio transmitter.
- Every vehicle is structured as illustrated in Figure 4.2, therefore, all information required from the application layer is available.
- Only highways are considered.
- There are no accidents on the roads.

4.2.2 Communication Requirements

The OTHA protocol must meet certain requirements defined for every communication systems employed in VANETs. Similarly, certain basic requirements must be met for the proper functioning of the OTHA protocol. In this work we consider the pre-requisites defined in the Car2Car Communication Consortium document published in [Baldessari *et al.* 2007], as listed in the following:

- **Entities utilized:** although expected to enhance the vehicular communication, *Vehicle to Road Site Unit*, *Road Site Unit to Road Site Unit*, and *Vehicle to Infrastructure* are not required or assumed in this work. The only entities utilized by the OTHA protocol are the vehicles, thus, *Vehicle to Vehicle* communication.
- **Communication type:** because the OTHA protocol relies on the geographical position of vehicles in order to disseminate the acquired traffic information upstream on the road, the communication type employed is referred to as a *directional broadcast*.
- **Transmission type:** the radio range provided by the wireless transmitters equipped into vehicles are required to be symmetric, i.e., if a vehicle C1 can communicate with a vehicle C2, a transmission from C2 will also reach C1.
- **Wireless radio technology:** vehicles must be equipped with a network device for short range wireless communications based on IEEE 802.11p radio technology.
- **Transmission power:** the vehicle must be able to constantly transmit at high power levels without battery constraints, as it often occur in Wireless Sensor Networks (WSN) [Akyildiz *et al.* 2002].
- **Frame size:** the OTHA protocol presented in this work does not support fragmentation, therefore, the amount of information exchanged is limited by the maximum payload size defined by the 802.11p protocol. The generic maximum payload size for 802.11 protocols is defined as 2312 bytes [Gast & Loukides 2002]. In case this values is exceeded, the upper layer Traffic Filter Protocol Layer will take care of discarding, compressing, and therefore, reducing the traffic information size to the value mentioned.
- **Anonymity and Data Security:** even though not provided by the OTHA protocol, the communication among vehicles has to be anonymous and secure. Anonymity is important to protect the identity of vehicle and drivers. Security plays a crucial role in preventing the insertion of false data into the network, which could result in situations as serious as accidents on the road.
- **Effective Protected Frequency Band:** There must be an exclusive frequency band destined to safety applications. It is unacceptable that other transmissions of lower priority, e.g., users downloading video on the Internet, interfere with critical safety applications. For instance, the OTHA protocol does not address critical situations such as accidents on the road and, thus, transmissions carried out by the protocol must not interfere with other safety applications.
- **Mandatory Sensor Data:** the following information must be provided by every vehicle:
 - Position data
 - Vehicle speed
 - Driving direction
 - Hazard warning signal flasher

- Brake power / vehicle deceleration
- ABS, ESP and ASR sensors
- Rain sensor / wiper status

These values are listed as obligatory in [Baldessari *et al.* 2007]. From the data list above the OTHA protocol requires the position data, vehicle speed and driving direction. In addition, as outlined in Chapter 3 the distance to the vehicle in front, current lane number of the vehicle, road identification (ID), location of junction points and junction point type are also required.

- **Scalability:** the OTHA protocol must work in situations with a very small density of road traffic and in situations with a very high traffic density, such as traffic jams or major intersections/junctions.

The Traffic Filter Protocol Layer

Contents

5.1	The TrafficMap	31
5.1.1	The TrafficMap structure	34
5.2	Traffic Filtering	35
5.3	The Traffic Filter Protocol Layer	39
5.3.1	Merging information	42
5.3.2	Adding an entry	45
5.3.3	Averaging	46
5.3.4	Reducing the TrafficMap	47
5.4	Compression Optimization	50
5.4.1	Greedy approach	52
5.4.2	Follow-lane approach	53
5.4.3	Analysis and comparison between the approaches	57
5.5	Summary of Parameters	61

In this chapter we describe in detail the upper Traffic Filter Protocol Layer.

The chapter is organized as follows: Section 5.1 describes the TrafficMap, the structure where all traffic information gathered from other vehicles downstream are stored and organized; Section 5.2 motivates the need for filtering part of the traffic information content; Section 5.3 provides details of the upper Traffic Filter Protocol Layer by giving an overview and describing each of its functions; Section 5.4 presents compression optimization methods for this layer; and finally, Section 5.5 provides a summary of all parameters utilized by this layer.

5.1 The TrafficMap

When traveling on the road, vehicles must receive accurate information regarding the traffic condition kilometers ahead up to the pre-defined *road segment*. Being the identification of traffic jams the main goal of the OTHA protocol, a speed profile of each lane on the road may suffice. This speed profile basically comprises the position and speed of vehicles which are located further ahead on the road, i.e., over-the-horizon. With this information available, traffic jams can be identified by evaluating the variation of vehicle speeds along the road.

In this work, the way in which such speed profiles are structured is referred to as **TrafficMap**, as defined by Van Eenennaam in [van Eenennaam 2008]. As the name implies, the TrafficMap

provides a map of the upcoming traffic. By means of **TrafficMap Messages** being exchanged among vehicles along the road in a distributed and ad-hoc manner, this map is constructed and updated periodically.

In his thesis, Van Eenennaam presents a TrafficMap meant to address single-lane roads. For this purpose, the TrafficMap regards the single lane present on the road and vehicles add a new entry (also referred to as sample) with their own position and speed upon the receipt of a new TrafficMap Message that contains information about the traffic condition ahead. When addressing multiple-lane roads, however, some new form of structure for the TrafficMap might be necessary to still provide accurate information about the current traffic condition of a road.

One straight-forward manner of extending the explained approach to multiple-lane roads is by having one TrafficMap separately for each lane. Figure 5.1 illustrates how these separate TrafficMaps are built along the road. In lane L_0 a vehicle constructs a TrafficMap by adding a new entry E_0 with its own current position and speed values. As TrafficMap Messages are received by vehicles upstream, new entries are added up to entry E_3 . Equivalently, vehicles in the remaining lanes have up-to-date information about their own lane.

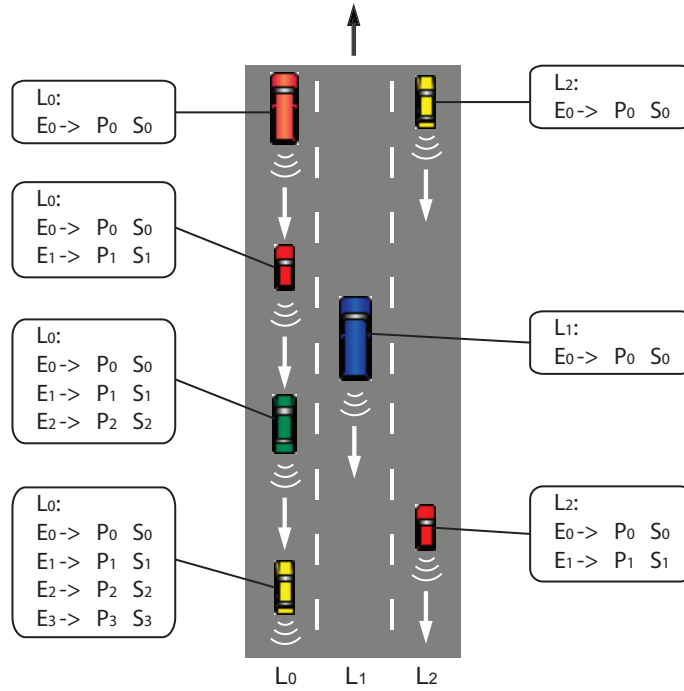


Figure 5.1: Separate TrafficMaps with the following notation: L (lane), E (entry), P (position) and S (speed)

Clearly, this first approach provides only a restricted view of the traffic ahead, since a vehicle would maintain only information regarding the lane it is currently situated in. In fact, if we assume for a moment that in this example the high vehicle density in lane L_0 indicates a traffic jam, all vehicles driving on lanes L_1 and L_2 would be unaware of it and consequently could decide to move to that lane, aggravating the traffic condition in that area. More importantly,

the requirements for the Congestion Assistant outlined in Chapter 3 state that a vehicle must have an accurate view of traffic jams that might be located in different lane(s) and even in different upcoming roads that a vehicle might take by means of junctions. The existence of several TrafficMaps and consequently of multiple *information traffic flows*, i.e., one for each lane on the road, also may not scale as numerous messages would be transmitted in parallel and greatly increase the utilization of wireless medium resources. Due to the mentioned drawbacks, this approach is not an option for the TrafficMap structure.

Another approach of dealing with multiple-lane road scenarios is by providing vehicles with information regarding every lane in a certain pre-defined road segment. In this way, every vehicle contributes to the construction of a common TrafficMap that is shared among all vehicles within the road segment by means of TrafficMap Messages. This unique shared TrafficMap could be thought of as a merging of individual TrafficMaps that concern specific lanes in a road. Figure 5.2 illustrates the building process of such TrafficMap. For the same example depicted previously, vehicles now have information about the downstream traffic for all lanes. A vehicle on lane L_0 creates the TrafficMap by adding its own information entry E_0 . A TrafficMap Message flows upstream and a vehicle on lane L_2 adds its own information entry E_0 . Note that each entry is now organized in a specific table structure determined for separate lanes. This process continues until a TrafficMap Message reaches the last vehicle upstream on the road, namely, the last vehicle in lane L_0 . This vehicle adds its own entry and the TrafficMap contains the complete overview of the upcoming traffic of the road considered in this example.

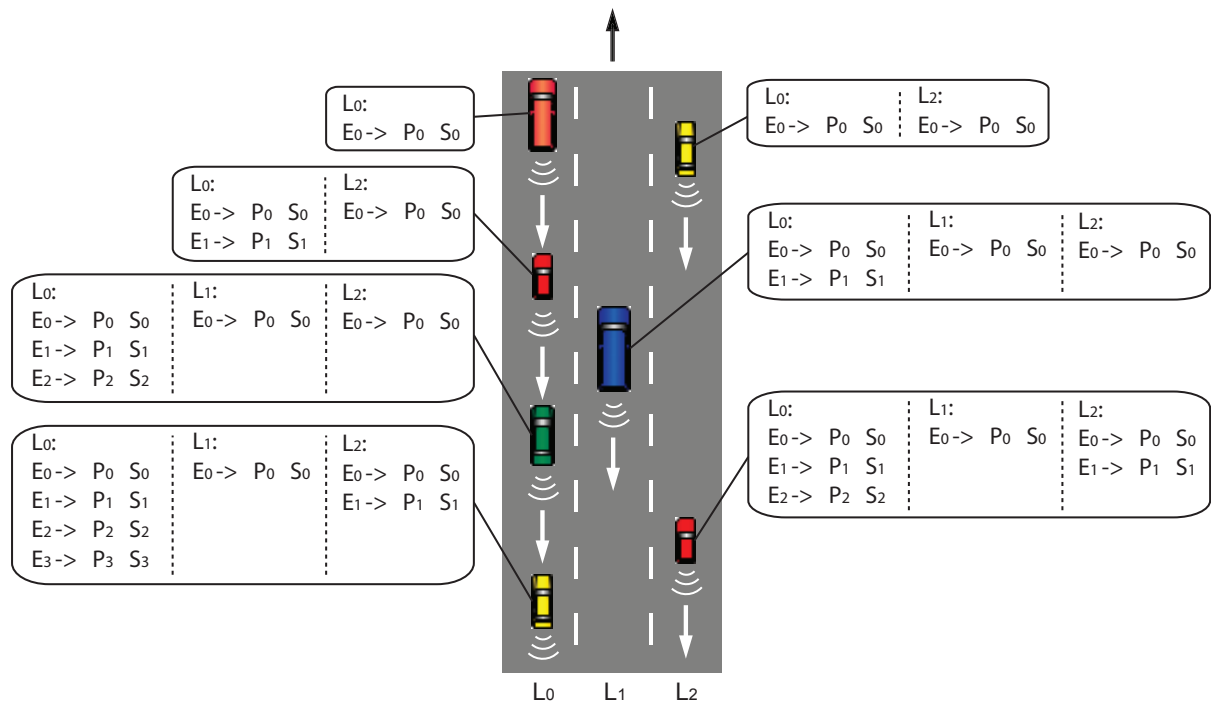


Figure 5.2: Merged TrafficMaps with the following notation: L (lane), E (entry), P (position) and S (speed)

Oppositely to the previous approach, a common TrafficMap with a complete over-the-horizon view indeed complies with the Congestion Assistant requirements and it is therefore preferable for the design of the TrafficMap structure. In the remainder of this work, such structure will be assumed.

5.1.1 The TrafficMap structure

A complete view of the TrafficMap structure is depicted in Figure 5.3. At the moment the TrafficMap is initiated, the *Flow initiator information* is set by the vehicle currently creating the structure. This information contains the *Road ID*, *Direction*, and *TimeStamp* values that together indicates the time and road where the TrafficMap has been generated. This extra information is utilized by vehicles to determine whether the structure received refers to a new information flow started on their own road. Such information flow serves as a means to provide control of a higher level overview of the traffic. In the description of the Traffic Filter Protocol Layer, we describe on which conditions the TrafficMap is required to be initiated. Furthermore, we will see that the *flow initiator information* also plays a role on the merging of information received from different TrafficMap Messages.

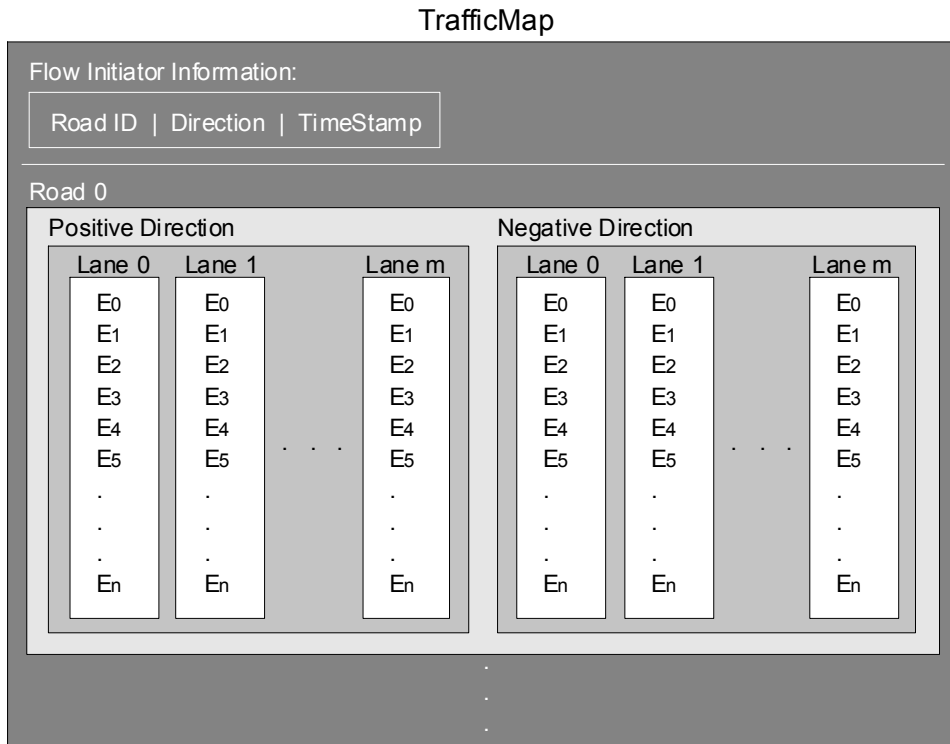


Figure 5.3: The TrafficMap Structure

The TrafficMap may contain information regarding one or more roads. Within a road there might be one or two directions, namely, the positive and negative directions. The entries added to the TrafficMap are finally organized in different lanes. Depending on the road, the number of

lanes will vary from one up to the number of lanes denoted as m of the widest existing highway [KatyFreeway 2009]. The number of entries per lane may vary from zero to a certain value n . This value starts from zero due to the fact there might be no vehicles in a certain lane during the time the TrafficMap is being spread on the road.

In order to offer the required information for the correct identification of the traffic ahead, each entry contains the following data:

- **Position:** x and y coordinates to identify the geographical location of the traffic jam.
- **Speed:** vehicle speed stored at the moment the entry was added to the TrafficMap.
- **TimeStamp:** time at which the entry was added to the TrafficMap. This value is important to determine the age of each entry and therefore it may be used to discard old entries.

5.2 Traffic Filtering

The identification of traffic jams required by the Congestion Assistant may be accomplished in different ways. One approach of providing the Congestion Assistant with such identification is by having all vehicles on the road segment adding their own information, namely, position and speed values. By analyzing the variation of vehicle speeds for different geographic positions, a decrease in speed for several vehicles in a certain area could indicate the occurrence of a traffic jam. Evidently, although this might offer the most accurate view of the traffic for some kilometers away, a new entry from every vehicle would increase the size of the TrafficMap enormously in wide roads with a high vehicle density. In fact, large message sizes together with high transmission power and high transmission rates have shown to be the main reasons for causing high radio channel congestion in vehicular network environments [Mittag *et al.* 2008]. Congestion in the radio channel is the result of overloading the wireless medium in a way the channel cannot support the amount of information traffic generated by vehicles. Especially when large messages are broadcasted, vehicles may often find the medium busy due to long transmissions in the network. This is sometimes referred to as the *unfairness problem* [Wischhof 2007], as the bandwidth available might be used by just a few vehicles sending out large messages and the network capacity is not shared fairly. A straight consequence of this behavior are the long delays in the information dissemination, which is not acceptable for a channel reserved for safety applications. Moreover, as supported in [Rezaei *et al.* 2007, Torrent-Moreno *et al.* 2005], enough bandwidth should be reserved for more delay-critical situations, such as accidents on the road.

Due to these reasons, it is important that the TrafficMap contains sufficient information to accurately represent the overview of the traffic ahead on the road and at the same time have its size limited in order to prevent high channel load levels. To achieve this goal, the information regarding the traffic ahead may be *filtered* to include only the necessary information that best represent the traffic condition ahead.

In fact, for the representation of traffic jams, the identification of position and speed of the head and tail vehicles in the congestion area would even suffice. However, determining precisely the head and tail vehicles of a traffic jam might be a difficult task, as finding a common traffic

pattern that accurately indicates whether a vehicle is leaving or entering a traffic jam without a complete knowledge of the network may not be always possible. For such indication, vehicles must be able to identify the occurrence of a traffic jam based only on local information. By relying on traffic flow theory, this could be achieved by means of traffic flow models. In [Artimy 2007], the local vehicle density is estimated based on Pipes' car-following model [Pipes 1953], the two-fluid model [Herman & Prigogine 1979] and the Nagel and Schreckenberg (NaSch) vehicle traffic model [Nagel & Schreckenberg 1992]. The relation between the local density and the speed of vehicles would indicate whether they are currently in congested or free-flow areas. Nevertheless, there has been some criticism over the models present in the literature, supported by the argument that there are still empirical observations that many traffic models do not reproduce [Schönhof & Helbing 2009]. In addition, the Congestion Assistant may also require more precise information about the speed profile of vehicles that are approaching the traffic jam in order to perform the active pedal (AP) function more accurately. As reviewed in Chapter 3, the AP function induces the driver to gradually reduce the vehicle's speed starting from a safe distance before reaching the tail of the traffic jam. This safe distance could be further improved if a braking curve of vehicles approaching the tail of the jam were available.

The traffic filtering could also be performed by having vehicles sharing summarized information regarding geographic areas to which they intend to drive or have already driven, as proposed in [Shibata *et al.* 2006]. By dividing the road map into fixed sub-regions called *areas*, each vehicle measures the time to pass through an area for each existing entering/exiting pair of roads of the area. Subsequently, each vehicle generates traffic information statistics based on the combination of its own information and the information received from vehicles which have passed through the same pair of roads previously. By doing this, drivers are able to learn the current congested areas and the estimated time required to get to their destinations. In this approach, the information contained in the messages exchanged among vehicles concerns only delay measurements for each possible route in a certain area. To prevent redundancy of information and a consequent growth in the message size, various mechanisms are proposed. Although this method may provide vehicles with an overview of expected delays when traversing congested areas, it does not meet the information requirements for the Congestion Assistant, as described in Chapter 3. As mentioned before, the Congestion Assistant requires detailed information regarding the position and speed of the head and tail vehicles of a traffic jam. Since these delay measurements are averages and mind entering/exiting pair of roads of an area, the estimated duration time of traversal in that area might not provide sufficient information to determine when exactly a traffic jam begins. Furthermore, clearly this approach does not offer a detailed speed profile of individual lanes on a road. Hence, the estimated average delay may mislead the driver when there are roads with a long traversal average delay but when in fact there could be some lane(s) with free-flow traffic. This could exist, for instance, in city entrances when roads may split into different paths to distribute the traffic to different parts of the city.

A rather simple approach is suggested in [van Eenennaam 2008]. A new entry containing the position and speed of a vehicle is added to the message being exchanged whenever it is considered important for the representation of the traffic ahead. The importance of samples is defined by thresholds that capture speed deviations of vehicles on the road. This approach is depicted in Figure 5.4 for a single-lane road. The speed variation of vehicles is modeled with the *Intelligent Driver Model (IDM)* [Treiber *et al.* 2000]. Although traffic-flow models might

not capture every scenario found on the roads, it still serves as a good indication of how vehicle speeds are related, based on common driver behaviors. In this example, a message is generated by the vehicle at the last position of the figure, closer to 10 km, and propagated to vehicles upstream. Upon the receipt of a message, vehicles evaluate the previous speed added to the message and compare it with their own speed. The difference between the speeds are evaluated by means of the pre-defined threshold value to verify whether the speed deviation is significant for a new entry to be added to the message. The result is that, for a certain observer vehicle at position zero, the message contains fundamental entries with the speed and position of a few vehicles that could be represented by *spots* (red points in the figure) that give an overview of the traffic ahead, up to 10 km away in this example. The threshold-based approach has shown to be sufficient to capture detailed information of traffic jams including estimates for the position and speed values of the head and tail vehicles, as studied in [van Eenennaam 2008]. Due to its simplicity and compliance with the Congestion Assistant requirements, this will be the approach considered in this work.

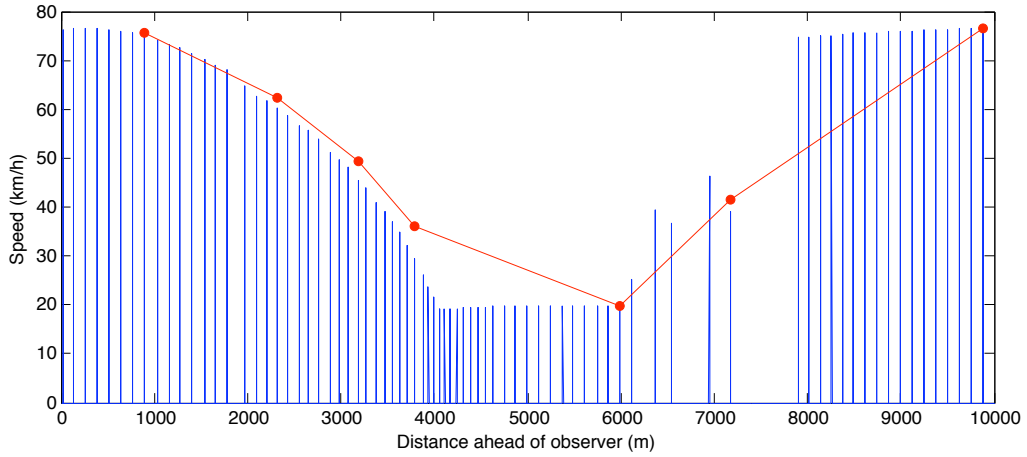


Figure 5.4: The threshold-based approach for a road with a single lane

When extended to a multiple-lane road, the threshold-based approach captures an overview of each lane of the road separately, as depicted in Figure 5.5. In this example, the decrease of vehicle speeds in the first two lanes indicates a traffic jam while the almost constant speed in lane three indicates free-flow traffic. This scenario may occur in situations when the road diverges into different paths, e.g., in a junction, and one alternative path formed by the right-most lane has a low traffic load of vehicles.

Another manner of representing the traffic condition of a certain road segment is by providing a top view of the road map, as shown in Figure 5.6. This might be useful when junctions have to be represented more accurately. The bars represent each lane and the arrows indicate the current lane direction on the road. The possible traffic conditions in this example are: free-flow traffic, moderate dense traffic, and traffic jam. In 5.6(a), a multiple-lane road is illustrated with some lanes varying their current traffic condition from moderate dense traffic to traffic jam. In 5.6(b), road R_1 with a single direction is connected with other roads (R_2 and R_3) by means of

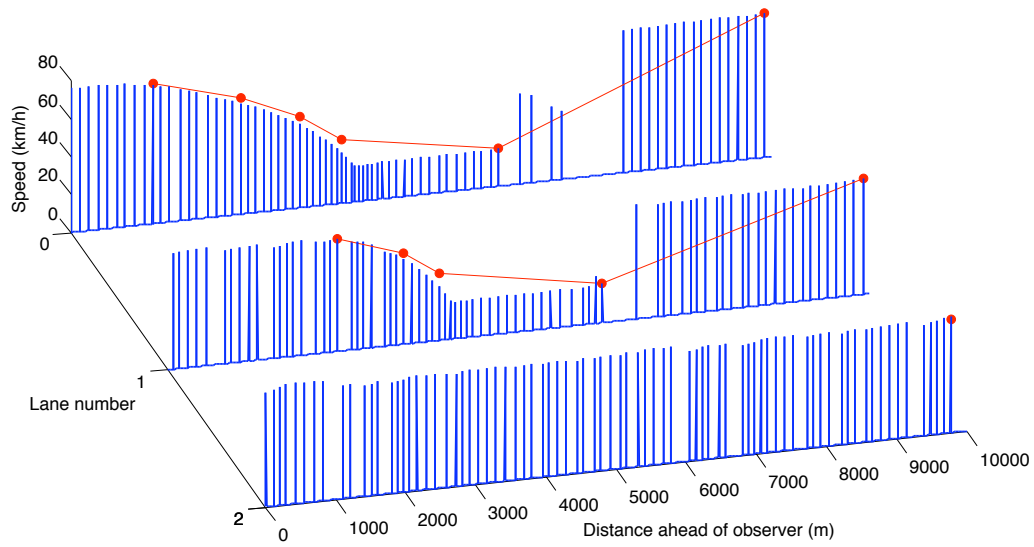


Figure 5.5: The threshold-based approach for a road with multiple lanes

a junction represented by a rectangle. In this scenario, there is a heavy traffic on roads R_2 and R_3 , while R_1 is currently indicating a free-flow traffic.

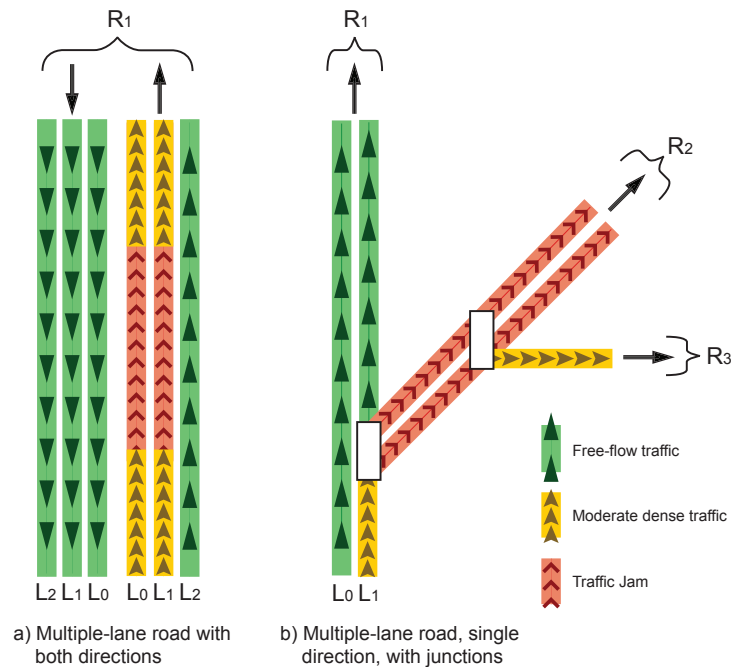


Figure 5.6: Top view representation of the road for different traffic conditions

5.3 The Traffic Filter Protocol Layer

Based on the decisions made regarding the content of the TrafficMap and the mechanism for filtering the amount of traffic information exchanged among vehicles, the Traffic Filter Protocol Layer is designed. As motivated in the previous sections, the TrafficMap will contain information of every lane for one or more roads. Moreover, the threshold-based approach will be employed as a manner to filter the traffic information.

To ease the understanding of the Traffic Filter Protocol Layer and other aspects of the OTHA protocol given in the remainder of this work, the following classification is applied to vehicles. Based on the threshold-based mechanism, vehicles can either add a new entry to the TrafficMap or simply relay the current information. Vehicles which add a new entry are referred to as *source vehicles*. When vehicles receive TrafficMap information and have no relevant information to add, they may relay the TrafficMap by means of TrafficMap Messages to other vehicles upstream. These vehicles are classified as *relay vehicles*. In addition, vehicles which are responsible for initiating a TrafficMap information flow, i.e., they are the first of a vehicle cluster and, therefore, are not able to receive up-to-date information from other vehicles on that road, are classified as *flow initiator vehicles*. Since they still add a new entry during the TrafficMap creation, they are a special type of source vehicles. Furthermore, the decision made upon whether or not the vehicle is a flow initiator depends on the current network connectivity in order to identify when a vehicle is the head of a vehicle cluster. As a result, such classification will be given by the lower *Dissemination Protocol layer*, as it is meant to address dissemination and networking issues of the OTHA protocol. For the sake of exemplification, in Figure 5.7 a flow initiator vehicle creates and adds new information to the TrafficMap and passes it to other vehicles upstream, which in turn either simply relay or add new entries to the TrafficMap.

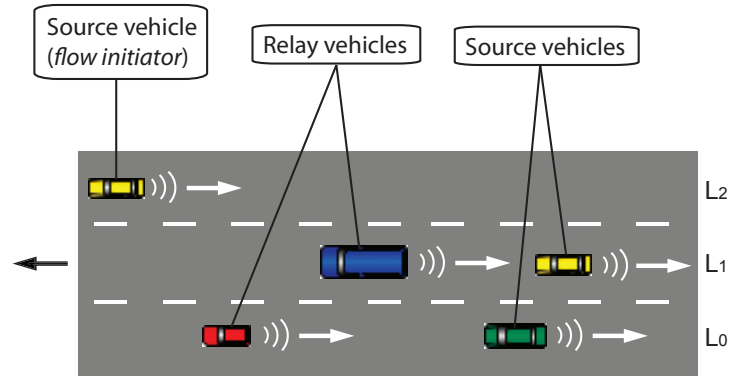


Figure 5.7: Example of vehicle classification

The *Traffic Filter Protocol Layer* functioning is depicted in Figure 5.8. Its main functionality is the *TrafficMap Manager*, which is a set of functions responsible for maintaining and updating the TrafficMap stored in vehicles. There are three possibilities for initiating the *TrafficMap Manager* process: upon the receipt of new TrafficMap information contained in a message originated by other vehicles and received from the lower layer; by the occurrence of pre-determined internal events within the vehicle; or by a TrafficMap data request received from the lower layer.

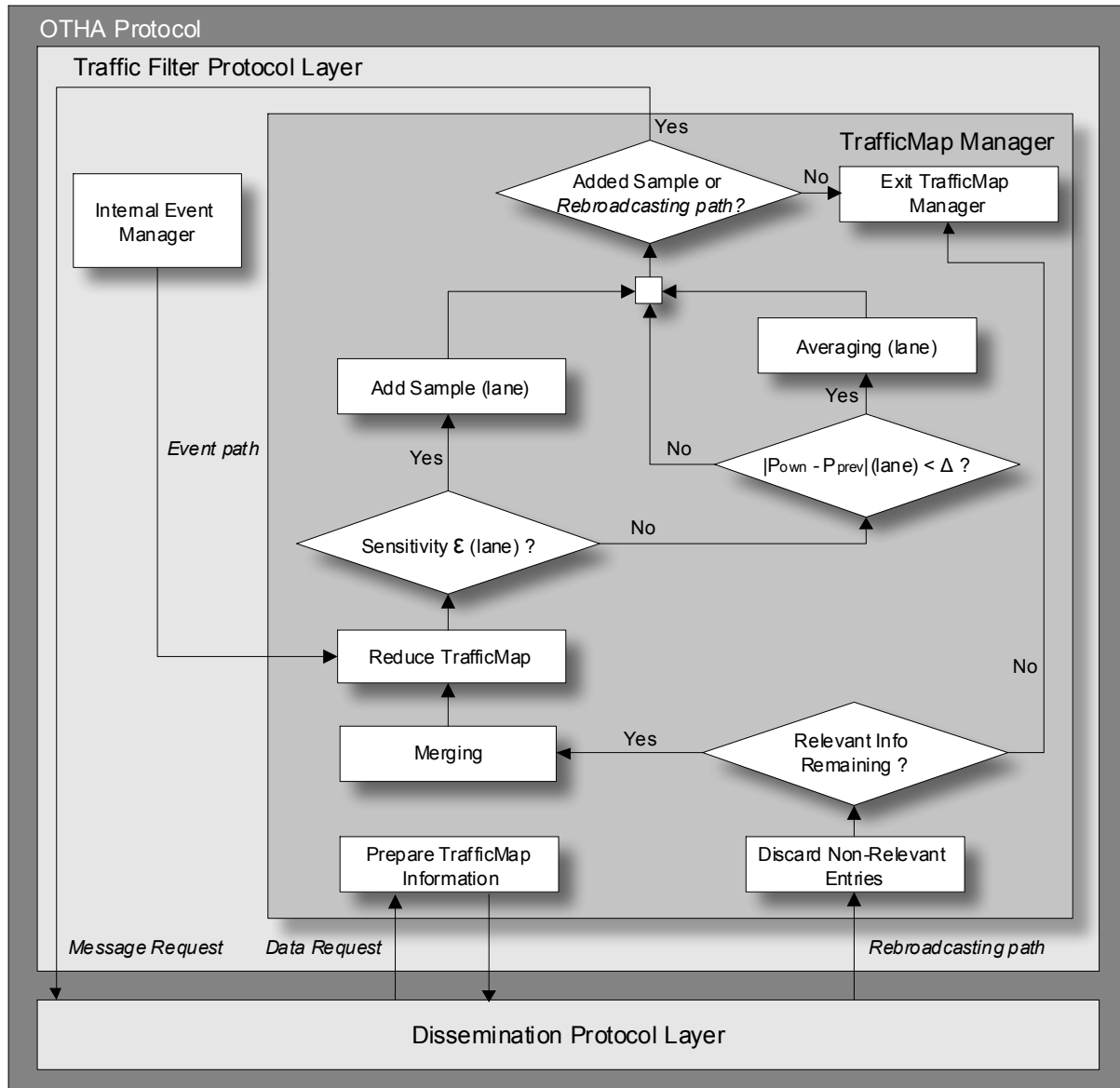


Figure 5.8: Overview of The Traffic Filter Protocol Layer

The former is meant to address the rebroadcasting of information regarding the speed profile of lanes on the road ahead. This information arrives from the lower layer by means of the *Rebroadcasting path*. From this path, the information received by other vehicles is first analyzed and all “non-relevant” information is discarded. The decision on which information is relevant or not will depend on the application running above the OTHA protocol, i.e., the Congestion Assistant. For the sake of simplicity, we define that roads where the vehicle is not able to go within the current *road segment* considered are not relevant. After this first filtering of information, if there is still some relevant information left, it will be *merged* with the current stored TrafficMap information. This process will keep the most up-to-date information regarding each lane of the roads considered. In order to reduce the current TrafficMap size, some old entries may be removed with the *reduce TrafficMap* function. The execution of this function at this point is especially important to guarantee that old entries containing position values behind the vehicle are removed before the execution of the next function, the *sensitivity* ε . The latter function decides whether a new entry must be added to the TrafficMap based on the last entry added to the lane on which the vehicle is currently situated. Therefore, the mentioned old entries would lead to wrong decisions and consequently in an inaccurate TrafficMap. Whenever the *sensitivity* ε decides not to add a new entry, the vehicle can still improve its TrafficMap by performing an *averaging* of its current speed with the last speed value received for its own lane. This averaging is only performed up to a certain distance from the vehicle which added the last entry, since very distant vehicles may not be representative for that entry anymore. Finally, a request is sent to the lower layer in order to disseminate the updated TrafficMap to other vehicles. As part of a rebroadcasting process, other vehicles upstream must also receive the updated TrafficMap even if no entry has been added. Due to this fact, the last decision step of the TrafficMap Manager will always allow the sending of the mentioned request to the lower layer. As there are decision processes in the lower layer that rely on the current vehicle type, the *message request* includes information about whether the vehicle is a source vehicle, i.e., it has added a new entry to the TrafficMap, or is simply a relay vehicle, i.e., it may have simply performed the averaging process.

Differently, the second form of initializing the TrafficMap Manager process concerns events that are triggered by the *Internal Event Manager* by means of the *Event Path*. In this path, the functions executed are basically the same when compared with the process initiated by the *Rebroadcasting path*. The exceptions are the *merging* and *discard non-relevant entries* processes, since now no information from other vehicles is received and, therefore, the execution of these functions is not necessary. Because the vehicle is not participating in a rebroadcast process, the last step in the TrafficMap Manager process will only permit a request to be sent to the lower layer for dissemination of the current information in situations when a new entry has been added to the TrafficMap. Otherwise, the process is simply finished.

In this work, the following internal events are considered:

- ***The periodical speed check/reduce TrafficMap timer has expired:*** this timer forces the vehicle to compare its current speed with the last speed value added to the TrafficMap for its current lane. When a sudden speed change occurs, the *sensitivity* ε function will allow a new entry to be included in the TrafficMap, which consequently will result in a new message request to warn vehicles behind about it. Since the event path starts by triggering

the *reduce TrafficMap function*, this timer also guarantees that old or entries which are no longer relevant are periodically removed even though the vehicle has not received any message from vehicles ahead.

- ***The vehicle has moved to another lane:*** whenever a vehicle moves to a different lane, its current speed might deviate considerably from the last speed value added to that lane. For such situations, a new entry will be added to the TrafficMap and a message request will be sent to the lower layer to warn vehicles upstream.
- ***The vehicle has moved to a different road:*** similarly to a lane change, when a vehicle moves to a different road, an evaluation regarding its current speed with the last entry added to the lane of the new road must also be carried out. Moreover, this will give the chance for the *Reduce TrafficMap* function to discard all previous entries concerning the previous road, since it might not be relevant any longer.

The latter form of triggering the TrafficMap Manager regards the receipt of a data request from the lower layer. The *Prepare TrafficMap Information* process depicted in Figure 5.9 ensures that the most up-to-date information is included in the TrafficMap Message just before it is sent to other vehicles by means of the lower layer. Since the lower layer is responsible for defining whether a vehicle will initiate a new TrafficMap flow, the data request contains information that indicates whether an up-to-date *flow initiator information* must be included into the TrafficMap Structure. In case such information must indeed be included, all current entries regarding the current vehicle's road are erased and a new entry together with an up-to-date *flow initiator information* are added to the TrafficMap. Finally, the whole data is retrieved and sent back to the lower layer. Vehicles which are not *flow initiators* simply jump to the last step to include the most up-to-date TrafficMap information and return it to the lower layer.

In the remainder of this section, we give a detailed explanation of the main functions executed by the TrafficMap Manager, namely, *merging*, *adding*, *averaging* and *reducing*.

5.3.1 Merging information

One of the main problems raised by the lack of synchronization during the exchanging of TrafficMap Messages among vehicles is how to combine the information coming from different vehicles in such a way that the TrafficMap keeps the most accurate view of the traffic ahead. When considering a single-lane road, as studied in [van Eenennaam 2008], the information exchanged regards simply a single lane for a determined road. In such cases, synchronization is not of great concern, since all TrafficMap Messages are related to the same lane and, consequently, it suffices to have vehicles replacing the existing TrafficMap information by the one received from vehicles downstream. Differently, when dealing with more realistic scenarios, the asynchronism of messages being sent by vehicles, for instance, located on different roads brings up the requirement for a procedure to *merge* and keep only the most up-to-date information received from different sources. This is evident in simple junction scenarios, where vehicles approaching an exit point could receive information both from the current road ahead or a different road given as an alternative route in the junction area. In fact, even in a simple case of a single road with multiple lanes, the merging of information might be required as we show in this section.

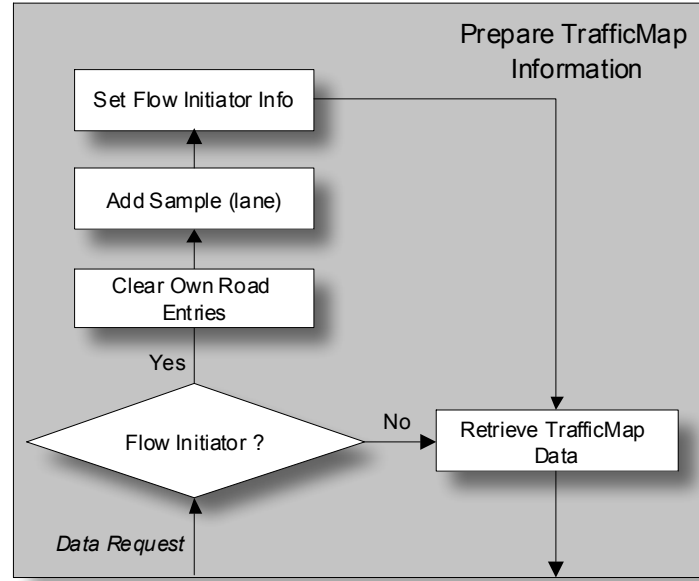


Figure 5.9: The Prepare TrafficMap Information Process

In order to always keep an up-to-date view of the traffic ahead, a simple solution is to rely on the use of *time stamps* that indicate the time of the last update for a specific lane. Upon the receipt of a new TrafficMap Message, the vehicle compares the time stamp of the last entry added to each lane with the time stamps previously stored in the local TrafficMap. If the last update for the information received regarding a certain lane has been performed more recently than what the time stamp previously stored for that lane indicates, all entries are replaced by the most up-to-date information. Because only information originated ahead the vehicle is relevant, at the moment of comparison entries added by the vehicle itself or other vehicles that now refer to geographical positions behind the vehicle due to the continuous mobility on the road are not taken into account during the merging process. In addition, whenever a vehicle receives a TrafficMap concerning a new flow for the same road it is driving on, i.e., it contains a more up-to-date *Flow initiator information*, all the existing local data for that road is replaced by the new one. This ensures that vehicles will keep the most up-to-date data for their own road. Since entries of a specific lane are only replaced when there is newer information received for that lane, in case there are no vehicles and therefore no entries for that lane during the new information flow, those entries would be kept and compromise the correct representation of the road. The time stamp of the new information flow guarantees that these entries are erased and no longer valid for the new flow received.

The basic merging process is illustrated in Figure 5.10 for a simple scenario of a multiple-lane road. At time t_0 , vehicle C_1 generates a new TrafficMap Message that contains a single entry E_0 regarding lane L_0 . Since vehicles C_2 and C_3 are placed within the transmission range of vehicle C_1 , they will receive and store the TrafficMap Message sent. At instance $t_0 + x_1$, both vehicles C_2 and C_3 have some information to add about their own lane, i.e., they are source vehicles.

At this moment, C_2 is the first to transmit a message with its own TrafficMap updated. As vehicle C_4 is now within the transmission range of C_2 , it will receive and store the TrafficMap information. The merging process occurs at instance $t_0 + x_1 + x_2$. The entry E_0 added by vehicle C_3 regarding lane L_1 is sent together with entry E_0 of lane L_0 previously received at time t_0 . Clearly, vehicle C_4 has already received the most up-to-date information for lane L_0 , namely, entries E_0 and E_1 at instance $t_0 + x_1$, and it will keep it. On the other hand, the information received for lane L_1 is certainly included, as no information regarding that lane had been received before. Consequently, by analyzing the time stamp of the last entries for each lane in the TrafficMap, only the most up-to-date information is stored by vehicles, which is desirable for maintaining the most accurate view of the traffic ahead.

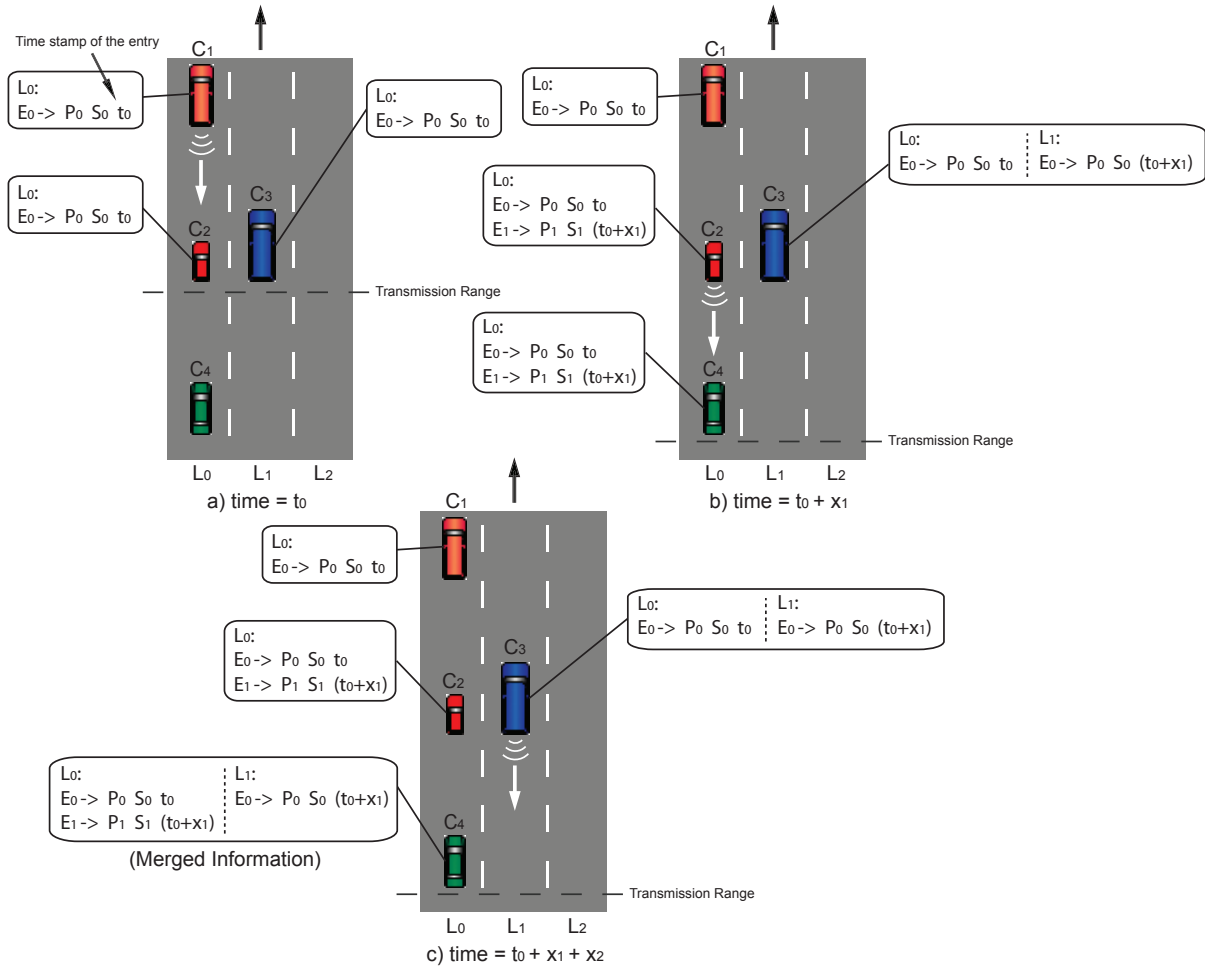


Figure 5.10: Example of TrafficMap information being merged

5.3.2 Adding an entry

During the TrafficMap formation, the *sensitivity* ε function is the process responsible for deciding whether a new entry must be added to the current TrafficMap. This decision is based on pre-defined thresholds that evaluate the distance and speed of vehicles. The former simply aims at keeping a *refresh* of the traffic ahead by forcing vehicles to add a new entry to their own lane in the TrafficMap whenever the distance limit d between the vehicle's position and the position value of last entry added to that lane has been exceeded, as follows:

if $|p_{own} - p_{previous}(l)| > d$ **then** add a new entry to lane l in the TrafficMap

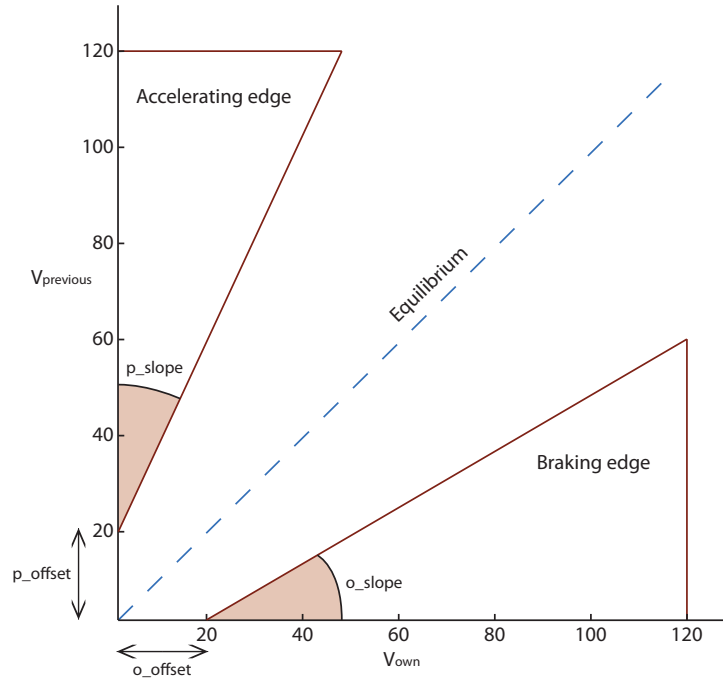
The latter threshold defines whether the difference between the current vehicle's speed and the previous speed value added to the TrafficMap for the current vehicle's lane is worth being considered as an important value for the representation of the traffic ahead. For a threshold ε and a current vehicle's lane l , the following decision is made:

if $|v_{own} - v_{previous}(l)| > \varepsilon$ **then** add a new entry to lane l in the TrafficMap

Clearly, different values for both thresholds influence directly on the number of entries added to the TrafficMap and thereby on the accuracy provided by the TrafficMap. While small values would result in an abundant number of entries added, great values would result in only a small set of points of the road segment that might not be sufficient to capture the current traffic behavior accurately.

An interesting manner of providing more flexibility to the speed threshold depending on whether the vehicle is increasing or decreasing its speed has been defined in [van Eenennaam 2008]. In the referred work, the *sensitivity* ε evaluates the speed deviation of vehicles as illustrated in Figure 5.11. By projecting the vehicle's own speed V_{own} and the previous speed value added to the current vehicle's lane $V_{previous}$, two areas are generated: the *braking* and *accelerating* edges. These areas are delimited by two slopes, namely, p_{slope} and o_{slope} . These slopes are the thresholds that determine whether an entry must be added to the TrafficMap. A vehicle will add a new entry to the TrafficMap whenever the relation between V_{own} and $V_{previous}$ matches one of the two areas. When the difference $|V_{own} - V_{previous}|$ is small, the points in the graph will be close to the equilibrium line. In such situations, there is only a negligible deviation between the speeds and no entry will be added. On the other hand, points falling within the defined areas capture important speed deviations and new entries are necessary to the TrafficMap. For instance, when a vehicle approaches a traffic jam, V_{own} will be greater than $V_{previous}$ and it might be worth adding a new entry to report this deviation. Remember that since the TrafficMap Messages are spread to vehicles upstream, the *previous* entry added to the TrafficMap regards the upcoming traffic (downstream). In case there is no previous entry for a considered lane, the vehicle will just add a new entry to the TrafficMap.

An interesting aspect to mention is that the p_{slope} and o_{slope} parameters permit an individual sensitivity adjustment for the braking and acceleration edges. In practice, since the Congestion Assistant requires more details regarding vehicles arriving in a traffic jam, the braking edge can be set as greater when compared to the acceleration edge. In this way, more entries would be generated from vehicles approaching a slower traffic ahead.

Figure 5.11: The threshold-based *sensitivity* ε function

Other parameters included in the *sensitivity* ε function are the p_{offset} and o_{offset} offsets. The reasoning behind them is that otherwise new entries would be constantly added to the TrafficMap when vehicles are traversing a traffic jam. In such situations vehicle speeds may vary constantly from zero to a certain slow speed value, e.g., 20 km/h. Nevertheless, the tail and head of a traffic jam are still captured by detecting the speed deviation of vehicles leaving or entering the traffic jam.

5.3.3 Averaging

An enhancement for the speed values of the entries included in the TrafficMap has been proposed in [van Eenennaam 2008]. As previously described, the *sensitivity* ε function aims at capturing speed deviations present in each lane of a road segment. The entries added to the TrafficMap, although representing fundamental points of the current state of each lane, can be further improved by letting other vehicles upstream perform an averaging of the last speed value added with their own (current) speeds, whenever the *sensitivity* ε function does not allow them to add a new entry. For each lane, vehicles upstream would update the last speed value by performing the mentioned averaging. The updated value is included in the TrafficMap Messages exchanged and other vehicles upstream in the same lane would repeat the process, until the *sensitivity* ε function detects a new entry to be added or when a certain pre-defined distance limit Δ is reached. By applying this averaging up to a certain distance defined by the parameter Δ , the entries would provide an enhanced local view of the general behavior of vehicles located at that region. For instance, if an entry is added by a vehicle during an overtake, the speed

value included into the TrafficMap might be an overestimation of the average speed of other vehicles behind that are in the same lane. The averaging process would then be responsible for improving and providing a better local estimation of this last value added.

Equation 5.1 defines how the average of an unknown number of previous values can be calculated. Upon the receipt of a TrafficMap Message, vehicles which are not allowed to add a new entry into the TrafficMap by the *sensitivity* ε function will look at the previous (last) speed value added in their lanes. The average result is given in the new speed value v_{new} , by calculating the weighted average of the previously updated speed value $v_{previous}$ with the own vehicle's speed v_{own} .

$$v_{new} = \frac{v_{previous} + (v_{own} \times \theta(d))}{1 + \theta(d)} \quad (5.1)$$

The function $\theta(d)$ provides the weight value to be employed, varying from 0 to 1, based on the distance d between the current vehicle's position and where the original entry has been added, as defined in Equation 5.2. The $\theta(d)$ function gives more weight to speed values of vehicles near the position where the original entry has been added, since vehicles close to each other are more likely to behave similarly. As explained, the parameter Δ limits the distance up to where vehicles can still contribute with the averaging process. This limit lies in the fact that very distant vehicles might not be representative for the contribution of the last entry added. Hence, the distance d is a value between 0 and Δ ($0 \leq d \leq \Delta$).

$$\theta(d) = \left(\frac{\Delta - d}{\Delta} \right)^\alpha \quad (5.2)$$

The α parameter in Equation 5.2 determines how the weight value given by $\theta(d)$ relates with the distance d , as illustrated in Figure 5.12. With $\alpha = 1$, the weight value is proportionally decreased as the distance increases. On the other hand, with $\alpha = 8$, the weight value is rapidly decreased for vehicles further away. The opposite behavior occurs with $\alpha = 0.125$, where the weight decreases slowly as with higher values for d . In order to define a proper value for α further research is still required, since it may depend on the current traffic density or other factors not studied in this work.

5.3.4 Reducing the TrafficMap

During the TrafficMap formation, the number of entries added to the TrafficMap might increase indefinitely if no form of size constraint is established. As defined in Section 4.1, the *road segment* already imposes a limit on how distant the information originated by a vehicle must travel and be considered by the Congestion Assistant. Therefore, one straight-forward manner of preventing an enormous growth in the TrafficMap size is by discarding all entries originated beyond the current road segment borders. A pre-defined TrafficMap size limit could complement the use of road segments to maintain the TrafficMap size up to a maximum value. In this way, vehicles would have a maximum number of entries allowed for their TrafficMaps. This is particularly important, since vehicles should exchange as little information as possible to avoid the congestion problems mentioned in Section 5.2. The *reduce TrafficMap* function uses a combination of both

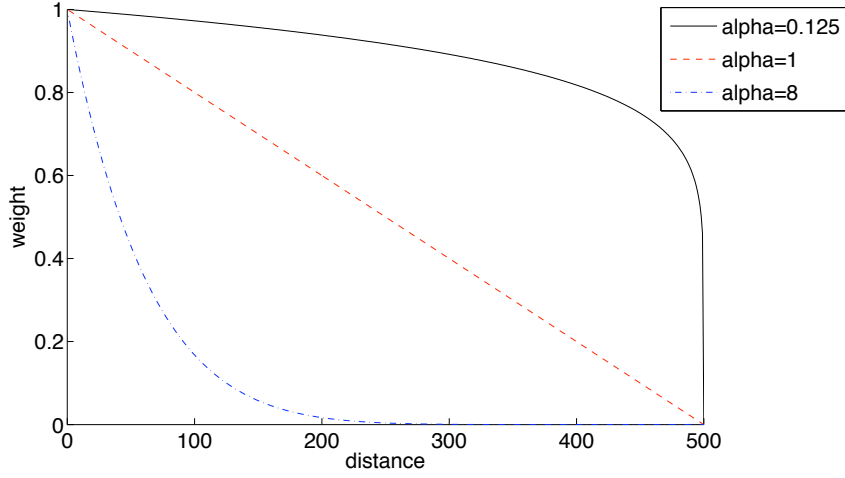


Figure 5.12: Weight θ as a function of distance d for different parameters of Δ and α

strategies to reduce the TrafficMap. When either there are entries beyond the road segment borders or the maximum number of entries has been reached, some entries will be discarded.

Since the road segment will normally comprise multiple roads with different traffic conditions, the process of discarding entries must be done wisely in order for the TrafficMap to still provide the Congestion Assistant with an accurate awareness of the upcoming traffic. Because remote information has a higher degree of uncertainty, entries regarding distant geographic positions should be preferred being discarded over entries with positions near the vehicle. Moreover, entries containing free-flow traffic information should be discarded sooner than entries with traffic jam information, as more detailed information may be needed for the Congestion Assistant to perform its functions more accurately when vehicles are approaching the congested area, as mentioned in Section 5.3.2.

Figure 5.13 depicts how both discarding rules can be employed. A top view of the current road segment abstracts the existence and exact position of vehicles by indicating the current traffic conditions for each road. In order to ease the understanding of the scenario considered, the observer vehicle C_1 used as reference is depicted. Vehicle C_1 drives towards a junction that connects two roads, namely, its own road R_1 and R_2 . In 5.13(a), a complete view of the traffic ahead is given when no reduction is performed. Assuming that the area considered does not exceed the pre-defined road segment but the number of entries needed to represent the complete area does exceed the limit pre-established, some entries must be discarded by vehicles downstream before the information arrives to C_1 . The result is that the TrafficMap arriving to C_1 will contain a reduced part of the road ahead, as marked with a darker line over the roads in 5.13(b). As illustrated, information regarding the traffic jam occurring in R_2 is maintained whereas some information of R_1 after the junction is not included.

It is worth noting that the number of entries needed to be included into the TrafficMap for each lane along the road will depend on the existence of vehicles and speed deviations in the traffic area considered. Due to the necessity for discarding entries from the TrafficMap, the distance up to which vehicles receive awareness is dependent on the current traffic conditions on

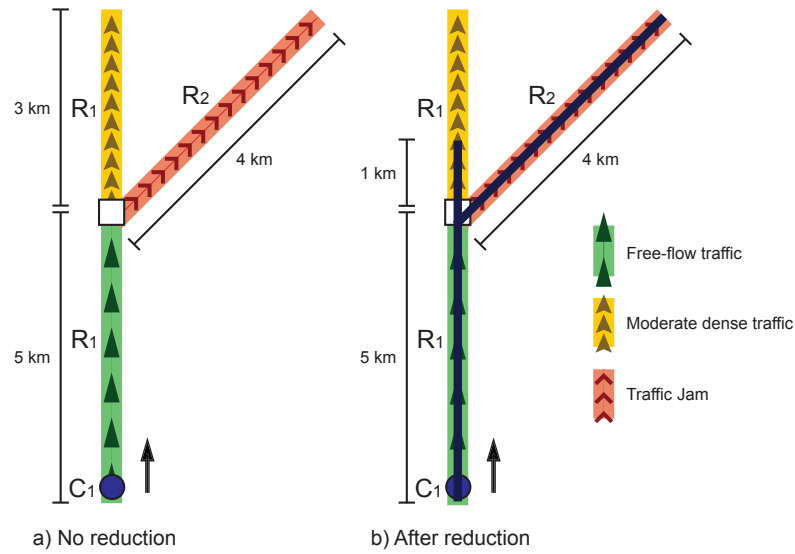


Figure 5.13: Example of reducing the TrafficMap based on the current situation of roads ahead

the road downstream. For situations where all vehicles are traveling on a single lane of a certain road, the total number of entries allowed to be used would regard that lane only. This would provide vehicles with more detailed information for that specific lane up to many kilometers ahead, probably limited by the road segment. Oppositely, in case there are vehicles traveling on all lanes of each road within the road segment and their speed varies constantly, the awareness provided to vehicles would be limited. The available total number of entries would have to be spread over the roads and since entries with near position values are preferred during the reduction process, the view of the traffic ahead would be more limited. Therefore, there is a limited number of entries that can be used to represent a certain road segment. They are used adaptively according to the current traffic state in that area.

Before meeting the described conditions for discarding entries from the TrafficMap, there might be some redundant entries that can be first removed. As suggested in [van Eenennaam 2008], two consecutive samples which are somewhat the same may be reduced to one. Furthermore, rapid increases and decreases may be summarized and reduced to fewer entries. In addition to these measures for reducing redundancy, vehicles must discard all entries (i) with position values behind the vehicle, (ii) from previous roads they have already passed by and thus are not relevant anymore or (iii) entries older than a certain parameter defined by the Congestion Assistant. The age of an entry is defined by the difference between the current time instant and the time stamp stored in each entry.

The summary of all measures taken to reduce the TrafficMap is listed in order of execution as follows:

1. Discard entries (i) with position values behind the vehicle, (ii) from previous roads or (iii) older than a certain parameter defined by the Congestion Assistant
2. Reduce redundancy:

- (a) Reduce two consecutive similar entries to one
 - (b) Summarize rapid increases and decreases
3. In case entries have been originated from an area that exceeds the pre-defined road segment or the number of entries included in the TrafficMap exceeds the limit pre-established, discard entries until both conditions are satisfied by following the priority rules:
- (a) Entries regarding remote geographic positions over positions near the vehicle
 - (b) Entries containing free-flow traffic information over traffic jam information

5.4 Compression Optimization

Although the *Reduce TrafficMap* function performed in the *Traffic Filter Protocol Layer* limits the TrafficMap size to a maximum value in a certain road segment, all information with respect to the entries discarded is completely lost. As motivated in Section 5.3.4, this may not be desirable when a great number of entries are required to represent a determined road segment, since any reduction could compromise the distance up to where vehicles receive an awareness. In fact, during a preliminary evaluation of the protocol by means of simulation, it has been found that a simple case of a road with two lanes can result in an average TrafficMap size of 487.18 bytes. When considering more complex scenarios with more lanes and roads this value is expected to increase rapidly to the maximum TrafficMap size permitted, which is in best case limited by the maximum frame size of the 802.11p protocol as described in Section 4.2 as 2312 bytes. Another approach for reducing the TrafficMap size is by compressing the existing information at the cost of some acceptable loss that are negligible for the correct representation of the upcoming traffic condition. In this way, more entries would be included in the TrafficMap and, thus, a longer distance of awareness of the upcoming traffic condition would be provided to vehicles. In this section, we propose two approaches for the TrafficMap compression that complement the *Reduce TrafficMap* function.

The basic compression principle adopted by the two approaches we describe is that vehicles in different lanes of a certain road and direction might have similar behavior at the moment the TrafficMap is being constructed. For instance, for traffic jams comprising multiple lanes, the difference between the speed profile of vehicles in the congested lanes might be negligible. For these lanes alike, any change in the speed profile for one lane may also be representative for the remaining lanes. Thus, the underlying compression idea is that a single entry can be used to represent speed changes occurring in different lanes, what could avoid considerably the insertion of new entries and an unnecessary increase in the TrafficMap size.

The compression mechanism is included in the *Add Sample* function in the *Traffic Filter Protocol Layer*. Whenever the *sensitivity* ε function detects a speed deviation between the current vehicle's speed and the previous entry in the current vehicle's lane, the compression algorithm will attempt to find in other lanes an entry similar to the one the vehicle is about to add. In case there is indeed a similar entry, a flag that indicates the entry is also valid for the current vehicle's lane is marked. Since this similar entry still remains in its original lane within the TrafficMap structure, a pointer referring to it may be added to the current vehicle's lane to keep the structure organized as depicted in Figure 5.3 (page 34). This pointer may serve as a

projection of the entry reused. By knowing the current lane width of the road, this projection may be accomplished by adding/subtracting a multiple number of the lane width to the entry's position value, as we show during the explanation of the compression algorithms. Notice that this pointer regards only the local TrafficMap representation in the vehicle's memory. For the TrafficMap Message encoding the flags added to the entries suffices for the correct TrafficMap reconstruction in the receiving vehicle. In addition, the functions *Merging* and *Reduce TrafficMap* are also slightly modified. Upon the removal of an entry containing marked flags, these functions will be responsible for managing the pointers for that entry. This could be done in different ways. For example, these pointers could be either removed or replaced by the content of the entry being removed.

The compression gain ratio for each entry is defined as follows in Equation 5.3. For a certain road and direction, l_{total} is the total number of lanes, l_c is the number of lanes with similar speed profiles that can be compressed, $entrySize$ is the total number of bits required to form an entry, and $flagSize$ is the total number of bits required to represent the flag used to indicate that the entry is also valid to another lane of the road. In this equation, the property $0 < l_c < l_{total}$ must hold, i.e., obviously there must be at least one lane to be compressed up to a maximum of $l_{total} - 1$, as one lane is to be followed.

$$compressionGain = 1 - \left\{ \frac{[(l_{total} - l_c) \times entrySize] + (l_c \times flagSize)}{l_{total} \times entrySize} \right\} \quad (5.3)$$

In order to motivate for the use of compression in the *Traffic Filter Protocol Layer*, let us assume a road with a total of 5 lanes, a flag size of 1 bit for each lane, and an entry size of 80 bits. The entry is formed by 8 bytes to represent the geographic position of the vehicle, as suggested in [Davis *et al.* 1996], and 2 bytes to represent the vehicle's speed in km/h. For these values, the compression of a single lane results in a gain of 19,75%, whereas the maximum compression possible with $l_c = l_{total} - 1 = 4$, gives a reduction of 79%. Clearly, since the number of bits required for each flag is negligible when compared with the entry size, any similarity in speed among vehicles in different lanes is worth the compression. Moreover, the maximum compression ratio possible increases as wider roads are considered.

The similarity between two entries in the TrafficMap from different lanes is verified by Algorithm 5.1. In contrast to how it has been used in the *Add Sample* function, the *sensitivity* ε is now employed to verify whether the speed values of two entries are similar, i.e., when *sensitivity* $\varepsilon = false$. The other value contained in each entry, namely, the position of vehicles, has also to be considered for the comparison of two entries. For this purpose, we limit the distance up to where two entries can be considered similar to each other. For a certain pair of entries being compared, the distance between their position values must be less than a pre-defined parameter d . The choice of d influences directly on the number of entries compressed during the TrafficMap formation. On the one hand, small values of d lead to more accurate comparisons and, thus, fewer entries compressed. On the other hand, great values of d results in a more tolerant similarity comparison and, hence, more entries compressed.

In the following, we describe the two compression approaches proposed to reduce the TrafficMap size, namely, the *Greedy* and *Follow-lane* algorithms. For both algorithms, we adopt the following variable notation:

Algorithm 5.1 $\text{similar}(p_{own}, v_{own}, p, v)$ - Similarity Verifier

```

1: if  $|p - p_{own}| < d$  and  $\text{sensitivity } \varepsilon(v_{own}, v) = \text{false}$  then
2:   return true
3: else
4:   return false

```

- l_{own} , p_{own} , and v_{own} contain the own vehicle's lane, position, and speed, respectively.
- Each entry variable e has three fields: $e.p$ represents the position value; $e.v$ represents the speed value; and array $e.lanes[]$ indicates whether other lanes have similar behavior to this entry, i.e., these are the flags explained previously. Thus, for a given lane $i \neq$ entry e 's lane, where $0 \leq i < l_{total}$, $e.lanes[i]$ can assume a value either *true* or *false*.
- For convention, position values, e.g., in p_{own} and $e.p$, increase towards vehicles located further downstream on the road. For instance, when a vehicle receives a TrafficMap Message from other vehicle further downstream, its own position value is smaller when compared to the sender position value.

5.4.1 Greedy approach

Greedy algorithms [Cormen *et al.* 2001] make the locally optimal choice at each stage with the hope of finding the global optimum for a certain problem. Although they may find less-than-optimal solutions for a vast variety of problems, they are fast, simple, and may work well for situations when a global knowledge is not available. This is especially true for the scenarios considered in this work, where the global vehicle topology is not known.

The *greedy* compression approach works as follows in Algorithm 5.2. Before adding a new entry to the current vehicle's lane, the algorithm searches for similar entries already present in the TrafficMap in different lanes (lines 1-4). The similarity is checked by means of the *similar* function. Only similar entries with positions values smaller than the position value contained in the last entry added to the own vehicle's lane are considered. This condition guarantees that the area implicitly defined by the *similar* function ($|p - p_{own}| < d$) does not overlap previous values added to the current vehicle's lane information. A list of similar entries is built and the entry with the closest position value to the current vehicle's position is chosen as the optimum value (line 6). In the sequel, the flag $\text{entryToReuse.lanes}(l_{own})$ that indicates the optimum entry which has been found will also be valid for the current vehicle's information is marked as *true* (line 7). In case no similar entries are found, a new entry is added with the own vehicle's information (line 9).

The functioning of the *greedy* approach is depicted in Figure 5.14. In 5.14(a), entries are added to the TrafficMap without compression starting at position 10 km and traveling upstream up to an observer vehicle at position zero. In 5.14(b), the *greedy* algorithm is performed for the same scenario, with parameter $d = 300$ meters. Notice that whenever there are similar entries within a region limited by d , the flag is marked and entries are reused. The points in the figure with surrounding circles represent the projection of the entries being reused in other lanes. The arrows point out the exact entries in which the flag was marked for each projection.

Algorithm 5.2 Greedy Compression

```

1: for each lane  $l \neq l_{own}$  do
2:   for each entry  $e$  from lane  $l$  and  $similar(p_{own}, v_{own}, e.p, e.v) = true$  do
3:     if  $e.p < elast.p$ , where  $elast$  is the last entry of lane  $l_{own}$  then
4:        $entryList \leftarrow e$ 
5:   if  $entryList \neq \emptyset$  then
6:      $entryToReuse \leftarrow$  entry  $\in entryList$  with closest position value to  $p_{own}$ 
7:      $entryToReuse.lanes[l_{own}] \leftarrow true$ 
8:   else
9:     add new entry with  $(p_{own}, v_{own})$  to  $l_{own}$  in the TrafficMap

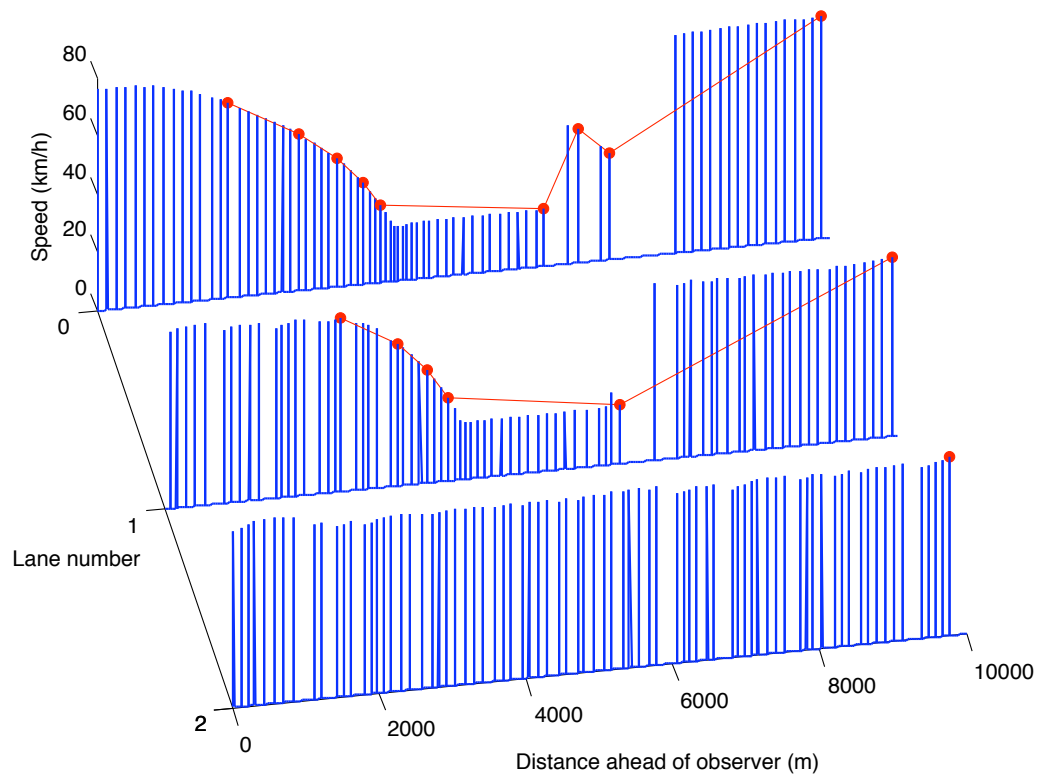
```

We can verify in this example that the TrafficMap size is considerably reduced and yet without preventing the TrafficMap from providing a representative view of the traffic ahead. In fact, from the 16 entries added without compression 6 entries are compressed, which results in almost 38% of compression efficiency.

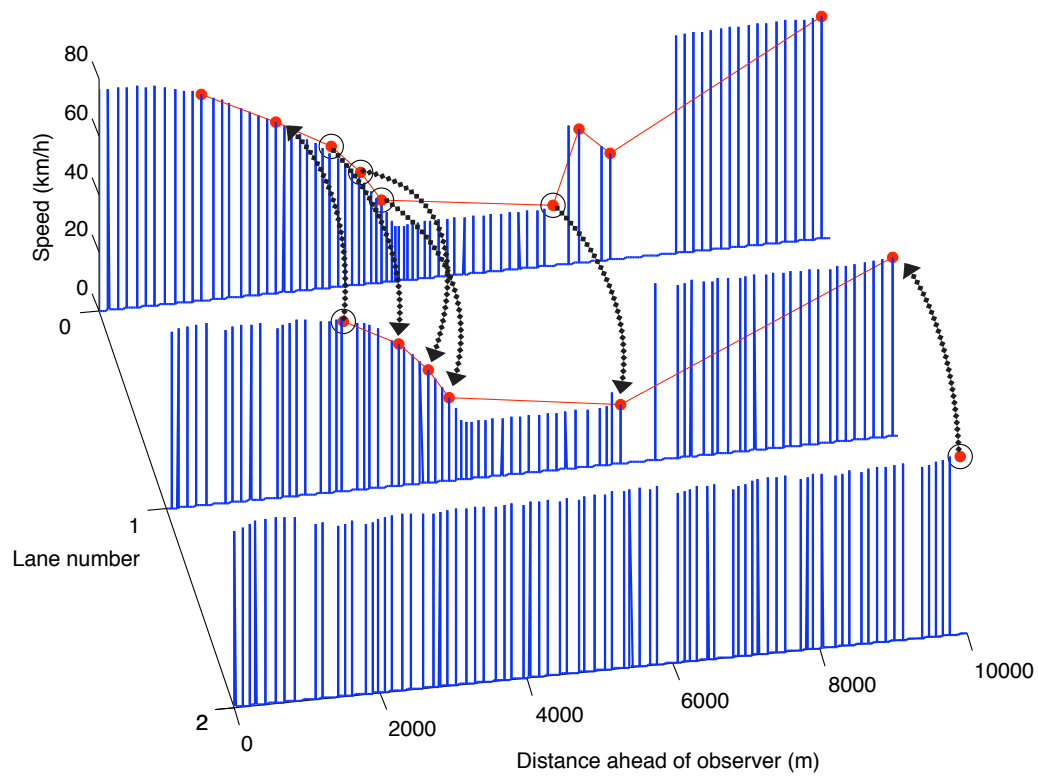
5.4.2 Follow-lane approach

During the exchange of TrafficMap Messages, vehicles might behave as *source*, *relay* or *flow initiator vehicles*, as described in Section 5.3. Because *source vehicles* or *flow initiator vehicles* (a special type of source vehicle described in Section 5.3) have important information to add, a new TrafficMap Message is always generated and sent to vehicles behind. *Relay vehicles*, on the other hand, simply help spreading the current information upstream. As we will see in Chapter 6, not every *relay vehicle* is required to rebroadcast the current information. If we consider, for instance, a group of *relay vehicles*, the furthest vehicle towards the message direction within the transmission range would suffice to relay the current TrafficMap. Therefore, if there are fewer vehicles acting as *source vehicles* on the road, fewer messages would be necessary to disseminate the current TrafficMap information. The compression approach we describe in this section aims exactly at diminishing the overall number of *source vehicles* on the road and at the same time it relies on the compression strategy of reusing similar entries in the TrafficMap whenever possible.

The *Follow-lane* approach relies on the assumption that when there are similar entries to be added on different lanes, these lanes might behave similarly not just for a single entry added to the TrafficMap, but also for the next few entries included along the road. In such situations, one or more lanes could *follow* another lane in a manner vehicles currently situated on *follower* lanes act as *relay vehicles*. When a vehicle on the lane which is being followed adds a new entry, it anticipates and marks the flag for the *follower* lanes indicating the entry is also valid for them. Ideally, this would avoid unnecessary TrafficMap Messages traveling on the road, since only vehicles driving on lanes being followed would act as *source vehicles*. Notice that in the *Traffic Filter Protocol Layer*, the decision of making a vehicle behave as a source or relay vehicle is directly associated with the *sensitivity* ε function. Therefore, we slightly alter this function to execute the *similar* algorithm whenever the vehicle is on a lane currently following another lane. If its current lane is still similar to the lane being followed, the *sensitivity* ε function returns *false*. Otherwise, the *Add Sample* is executed to either add a new entry or reuse an entry from



(a) No compression



(b) Greedy approach

Figure 5.14: Comparison between no compression and the Greedy approach

another similar lane to follow.

The total number of messages saved by having more vehicles acting as *relay vehicles* will depend on various factors such as the topology and the dissemination strategy adopted. For the sake of simplification, we say that whenever an entry marked with the flag as being valid to other lanes is in fact considered as valid by relay vehicles executing the *sensitivity* ε function on that lanes, one messages has *possibly* been saved. More specifically, given a certain followed lane l and a follower lane l_f , the number of *possible* messages saved includes all entries that have been compressed during the period l has been followed by l_f with the exception of the first and last entries compressed, i.e., when lane l_f starts and stops following lane l . The idea is that only vehicles driving on l_f can report in the TrafficMap that such lane has started following lane l . Similarly, l_f will stop following l when the current speed and position of the current vehicle driving on it are not *similar* to the last entry added by l . Thus, oppositely to the *Greedy Approach*, long sequences of entries being followed by other lanes are preferred over always trying to find the optimal compression.

In order for vehicles to know whether their current lane is a *follower* or *followed* lane, the following structure is included within the TrafficMap Message:

- The array *following* indicates the lane a given lane l is currently following and the position when it started to follow. The lane being followed and position values mentioned are indicated by *following* $[i].l$ and *following* $[i].p$ respectively, where $0 \leq i < l_{total}$.

The complete functioning of the *Follow-lane* approach is detailed in Algorithm 5.3. A lane can either be *followed* by one or more lanes or *follow* a certain lane. A vehicle first checks if the lane it is currently situated l_{own} is being followed by other lanes (line 1). The distance up to where a lane can follow another lane is limited by the parameter Ω . This decreases the likelihood of having vehicles driving on lanes being followed marking the flag in new entries for follower lanes indefinitely and, consequently, prevents situations where there are no vehicles (gaps) in certain follower lanes but entries are still marked as being valid also for them. To avoid this undesirable inaccuracy, all follower lanes that have reached their distance limit are removed from the *following* array (lines 2-4). A lane being followed can still look for some other lane to follow in order to enhance the current compression, i.e., having more follower lanes increases the compression since more entries are reused. However, because moving to another lane means taking all current follower lanes to the new lane, we define a more conservative approach that allows a lane being followed to follow another lane only if the last ϕ entries between them are *similar* (lines 5-13). Ideally, this would give more certainty regarding the similarity between these lanes, and avoid situations where lanes currently *similar* and following each other start following another lane but soon stop following it because their similarity is not valid any longer. In case the requirements for start following another lane is not met, the lane being followed simply continue adding new entries and marking the flag for the followers (lines 14-17).

The second part of the algorithm regards the case when a vehicle is situated on a lane that is currently not being followed by any other lane (line 18). There are two possible cases: the lane the vehicle is driving on is not being followed and not following any other lane, or the lane is currently following some lane and it will stop following because they are not *similar* anymore. In both cases, the algorithm tries to find a new similar lane to follow (lines 21-23). If there

exist indeed similar lanes on the road, the current lane will choose the one with higher degree of similarity and start following it (lines 24-28). Otherwise, it will simply add a new entry to the TrafficMap (lines 29-32). A special procedure is executed in case the lane has just stopped following some other lane (lines 19-20). The last entry added to the TrafficMap by some vehicle in the lane previously being followed is removed, since it has been inaccurately included to the TrafficMap.

Algorithm 5.3 Follow-lane Compression

```

1: if  $l_{own}$  is currently being followed by other lane(s) then
2:   for each lane  $l$  with  $following[l].l = l_{own}$  do
3:     if  $|following[l].p - p_{own}| > \Omega$  then
4:        $following[l] \leftarrow \emptyset$ 
5:   for each lane  $l \neq l_{own}$  and  $following[l] = \emptyset$  do
6:     if the last  $\phi$  entries of  $l_{own}$  and  $l$  are similar then
7:        $laneList \leftarrow l$ 
8:   if  $laneList \neq \emptyset$  then
9:      $laneToFollow \leftarrow \text{lane} \in laneList$  with closest position values to  $l_{own}$ 
10:    for each lane  $l$  with  $following[l].l = l_{own}$  or  $l = l_{own}$  do
11:       $following[l].l \leftarrow laneToFollow$ 
12:       $following[l].p \leftarrow p_{own}$ 
13:       $e.lanes[l] \leftarrow true$ , where  $e$  is the last entry of  $laneToFollow$ 
14:   else
15:     add new entry  $e$  with  $(p_{own}, v_{own})$  to  $l_{own}$  in the TrafficMap
16:     for each lane  $l$  with  $following[l].l = l_{own}$  do
17:        $e.lanes[l] \leftarrow true$ 
18:   else
19:     if  $l_{own}$  is currently following a certain lane  $l_{followed}$  then
20:        $e.lanes[l_{own}] \leftarrow false$ , where  $e$  is the last entry of  $l_{followed}$ 
21:     for each lane  $l \neq l_{own}$  and  $following[l] = \emptyset$  do
22:       if for the last entry  $e$  of  $l$   $similar(p_{own}, v_{own}, e.p, e.v) = true$  then
23:          $laneList \leftarrow l$ 
24:     if  $laneList \neq \emptyset$  then
25:        $laneToFollow \leftarrow \text{lane} \in laneList$  with closest position values to  $l_{own}$ 
26:        $e.lanes[l_{own}] \leftarrow true$ , where  $e$  is the last entry of  $laneToFollow$ 
27:        $following[l_{own}].l \leftarrow laneToFollow$ 
28:        $following[l_{own}].p \leftarrow p_{own}$ 
29:     else
30:       if sensitivity  $\varepsilon(l_{own})$  then
31:         add new entry with  $(p_{own}, v_{own})$  to  $l_{own}$  in the TrafficMap
32:        $following[l_{own}] \leftarrow \emptyset$ 

```

Figure 5.15 illustrates the Follow-lane approach being executed with parameters $d = 300$ meters and $\phi = 2$. At the beginning of the protocol execution at position 10 km, the vehicles driving on lanes 1-3 find the existing entry added to lane zero similar to what they are about to

add and start following it by compressing their entries. Between positions 6-7 km, lanes 2 and 3 stop following lane zero, since the *sensitivity* ε function performed by the vehicles on these lanes detects a speed deviation worth being reported. During their attempt to find other lanes to follow, vehicles in lane 3 finds lane 2 as an alternative and start following it. After vehicles in lane 2 realize that the last *phi* entries added to their lane were similar to lane zero, lane 2 starts following lane zero and brings its follower, namely, lane 3 with it. The compression is considerably improved and the similarity between all four lanes continues until vehicles in lane 2 detects another speed deviation. The process is finished by having lane 3 following lane 2 once again in the last entry added by them. For this scenario, 22 out of 34 entries are compressed resulting in a compression efficiency of $\sim 64\%$. The number of possible messages saved is 15. The actual percentage of saved messages over the total number of messages depends on the current radio channel condition, transmission range, among other factors, and it is left for future evaluation.

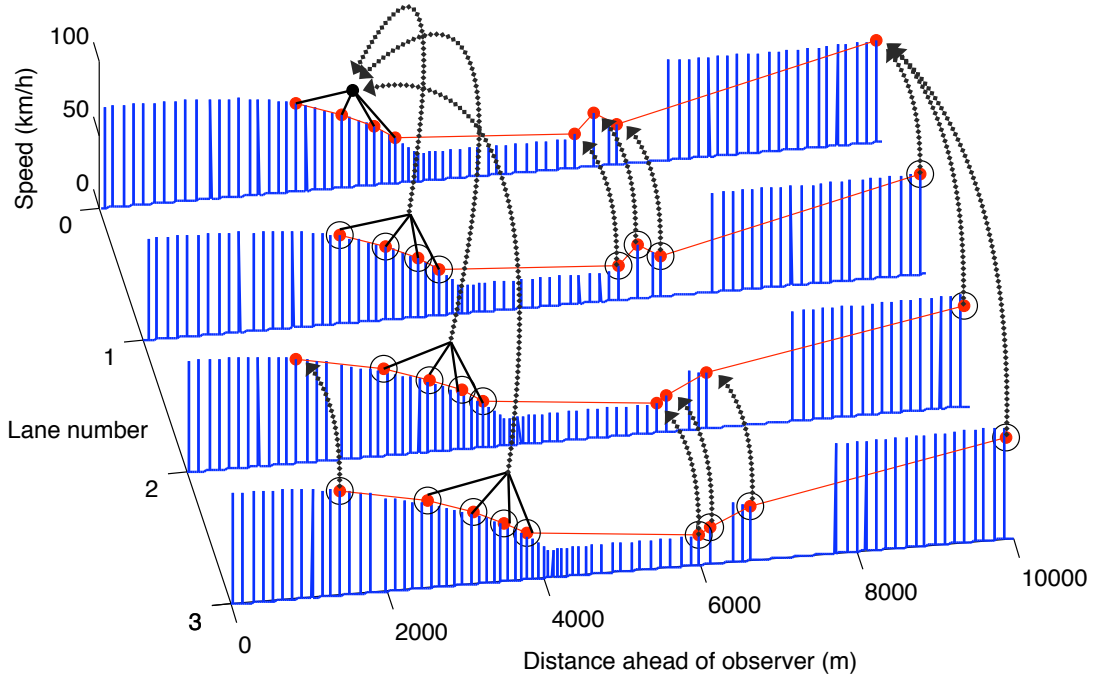


Figure 5.15: Example of the Follow-lane approach being employed

5.4.3 Analysis and comparison between the approaches

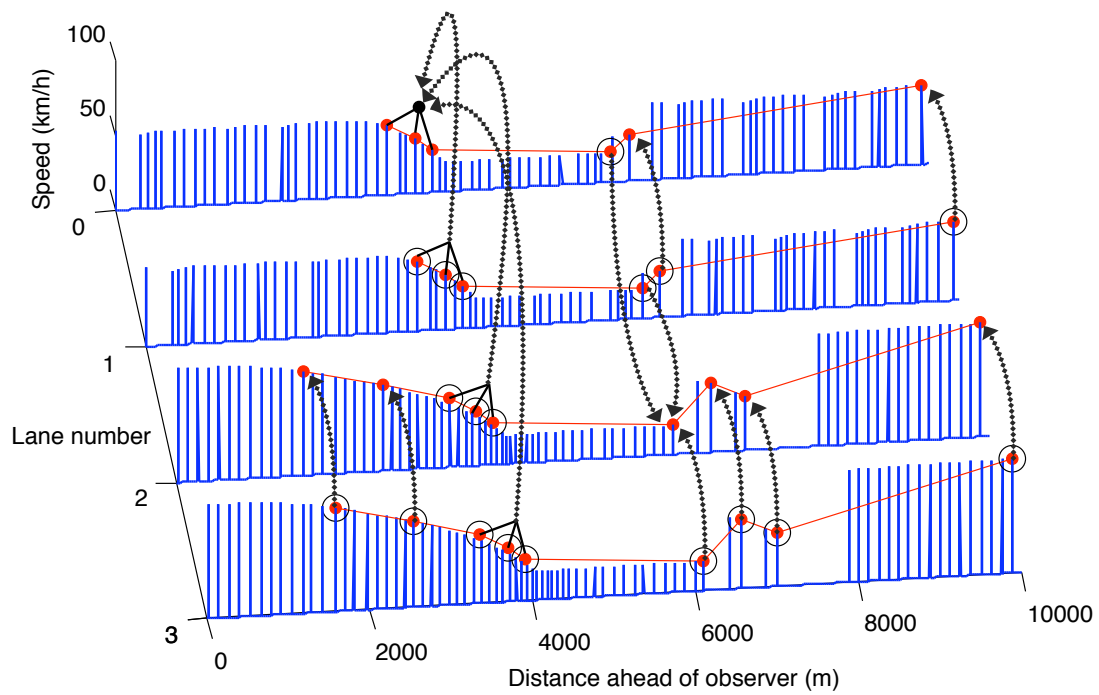
As we have seen, the compression approaches presented in the previous sections aim at different goals. On the one hand, the *Greedy approach* seeks the optimal local entry hoping to achieve the global optimum compression, i.e., the highest compression efficiency with the highest degree of similarity. On the other hand, the *Follow-lane approach* aims at reducing the total number of messages exchanged by having long sequences of compressed entries in lanes being followed instead of finding the local optimal compression.

The difference between both approaches is illustrated in Figure 5.16. For a given road with 4 lanes and parameter $d = 300$ meters, the behavior and goal of the two approaches is evident in this example. In 5.16(a), the *Greedy approach* always looks for the optimum entries when performing the compression, which results in entries from different lanes being reused randomly. Differently, 5.16(b) shows the *Follow-lane approach* being performed with parameter $\phi = 10$. In this example, two pairs of lanes following one another is evident and although it does not provide the best compression possible, the number of messages exchanged is reduced thanks to the long sequences of entries being reused by the follower lanes.

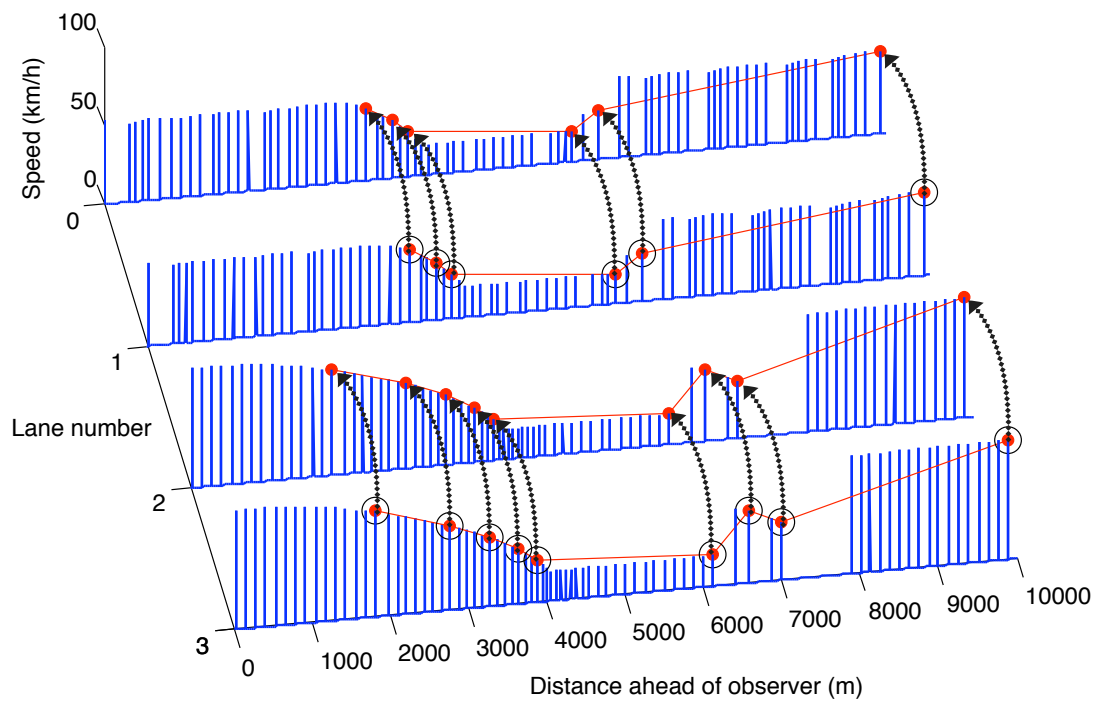
Clearly, the parameter ϕ influences directly on the quality of compression achieved by the *Follow-lane* algorithm. Great values of ϕ provides a more conservative behavior in order to decrease the number of messages exchanged and avoid situations where lanes are constantly choosing other lanes to follow to improve performance but at the cost of inserting more messages on the radio channel. Since the traffic behavior is often unpredictable, this value could be defined adaptively depending on data history that could be stored in the vehicle. Nevertheless, finding a proper value for ϕ requires further evaluation and it is out of scope of this work.

Another parameter utilized by the *Follow-lane approach* is the Ω value. This parameter addresses gaps that could often be present in some lanes of the road. Since the presence of vehicles is required to start or stop following another lane, gaps could severely damage the representation of that lane in the TrafficMap, as lanes being followed only stop marking the flag for the followers when the distance defined by the parameter Ω is reached. Lanes used for overtaking are clear examples of lanes that could present gaps of vehicles constantly, since they are used only when necessary. Therefore, by defining lower values of Ω , there would be shorter sequences of entries being reused by follower lanes. Oppositely, great values would permit the situation where a long gap of vehicles is present and, consequently, a higher inaccuracy in the TrafficMap would occur. Figure 5.17 illustrates an example of the *Follow-lane approach* being executed on the existence of a large gap of vehicles in lane one. Note that even though there is no great speed deviations on lane one, the entries added by vehicles in the followed lane zero indicate otherwise. Instead of using the Ω parameter, an estimate of the local vehicle density could tell whether there is a gap surrounding vehicles and therefore vehicles on lanes presenting gaps could cease the following of other lanes. However, deriving the local density often means utilizing more messages as shown in [Mittag *et al.* 2008, Tonguz *et al.* 2007], which is completely against our main goal.

Table 5.1 summarizes and compares the main characteristics of both approaches with respect to *number of parameters utilized*, *accuracy*, and *overhead*. Although the *Follow-lane approach* offers the opportunity of reducing the number of messages used to disseminate the TrafficMap information upstream, it is more complex, i.e., three parameters are required against one for the *Greedy approach*, it requires an extra overhead to control which lanes are following or being followed, and more importantly, it is susceptible to providing a completely inaccurate view of the traffic ahead in situations where gaps of vehicles are present, as shown in Figure 5.17. Due to the simplicity offered by the *Greedy approach* and the difficulty of solving the inaccuracy caused by the mentioned *gap problem*, this approach is chosen as the compression mechanism to be employed in the *Traffic Filter Protocol Layer*.



(a) Greedy approach



(b) Follow-lane approach

Figure 5.16: Illustration of the difference between both compression approaches

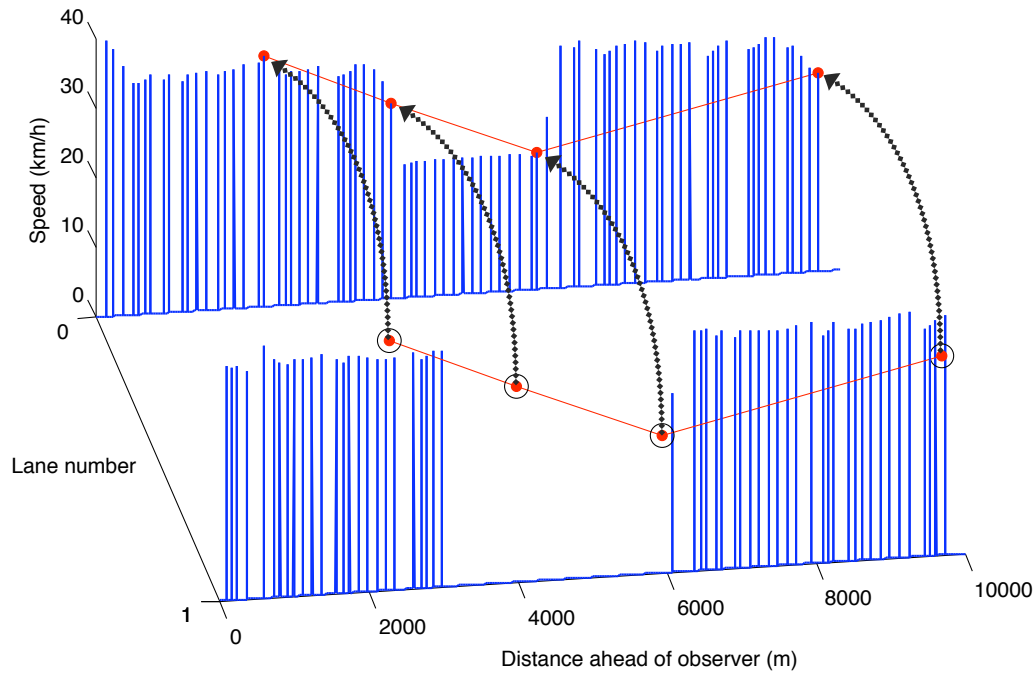


Figure 5.17: Exemplification of the gap problem in an execution of the Follow-lane algorithm

	<i>Greedy Approach</i>	<i>Follow-lane Approach</i>
Parameters	d	d , ϕ , and Ω
Accuracy	Chooses the local optimal entry in the TrafficMap. Accuracy is limited by the <i>similar</i> function.	Prefers long sequences of compression rather than optimal local entries. Gaps might severely compromise the accuracy depending on the Ω chosen and the current traffic condition.
Overhead	lane flags (1 bit p/ lane)	lane flags (1 bit p/ lane) and <i>following</i> array (8 bytes for position + 1 byte to indicate the lane it is following = 9 bytes p/ lane)

Table 5.1: Comparison between the compression approaches

5.5 Summary of Parameters

<i>Function</i>	<i>Parameter</i>	<i>Description</i>
Averaging	Δ	averaging interval
Averaging	α	averaging slope
Sensitivity ε	d	distance threshold
Sensitivity ε	o_{slope}	speed threshold for braking edge
Sensitivity ε	p_{slope}	speed threshold for accelerating edge
Sensitivity ε	o_{offset}	trigger-free zone for braking edge
Sensitivity ε	p_{offset}	trigger-free zone for accelerating edge
Reduce TrafficMap	road segment	maximum distance up to where the awareness is provided, defined by a geographical area.
Similar (compression methods)	d	distance threshold to consider two entries as <i>similar</i>
Follow-lane approach	ϕ	minimum number of similar entries necessary for a lane being followed to start follow another lane
Follow-lane approach	Ω	maximum distance a lane can be followed

Table 5.2: Parameters defined in the Traffic Filter Protocol Layer

The Dissemination Protocol Layer

Contents

6.1	Background	63
6.2	Addressing Source Vehicles	67
6.3	The Dissemination Protocol Layer	69
6.3.1	Time Manager	71
6.3.2	Message Builder	78
6.4	Time Slot Optimization	83
6.5	Summary of Parameters	85

In this chapter we describe in detail the lower Dissemination Protocol Layer.

The chapter is organized as follows: Section 6.1 provides background information and motivates for the use of a broadcast suppression mechanism; Section 6.2 describes the necessary modifications to a broadcast suppression mechanism when considering source vehicles; Section 6.3 provides details of the lower Dissemination Protocol Layer by giving an overview and describing each of its functions; Section 6.4 presents a time slot optimization method for this layer; and finally, Section 6.5 provides a summary of all parameters utilized by this layer.

6.1 Background

Due to the common asynchronous, dynamic and distributed communication environment found in VANETs, vehicles mostly rely on the broadcasting of messages in order to disseminate information to other vehicles. Establishing temporary routes in the network and relying on unicast communication is often not considered as an option, since the high mobility of nodes would result in severe high delay of route maintenance and set-up. In addition, numerous applications in VANETs disseminate traffic information that is of concern to most nodes on the road. For these reasons, our dissemination approach relies on the broadcasting of information to every node it concerns, upstream on the road.

Already known from previous studies in Mobile Ad-hoc Networks (MANETs), when blindly used, broadcasting messages might result in high *redundancy*, *contention*, and *collision* [Ni *et al.* 1999]. This is evident when performing the straight-forward approach that is referred to as *flooding*. In this mechanism, every node is responsible for rebroadcasting upon receipt of a new message. The redundancy is increased rapidly, since all nodes in the neighborhood will rebroadcast the same message and thus such message will be overheard several

times. In high density scenarios, the high number of messages rebroadcasted provokes an enormous contention period when nodes attempt to transmit due to the constant occupied medium. Finally, collisions might occur in case the medium is free and all nodes begin the rebroadcast process simultaneously. These consequences altogether are often referred to as the *broadcast storm problem*.

Several *suppression* techniques have been proposed in the literature in order to cope with the broadcast storm problem in MANETs. The prime goal is to have the minimum number of nodes rebroadcasting while still achieving a high penetration rate. In [Ni *et al.* 1999], a probabilistic and a few threshold-based techniques are proposed: the probabilistic, counter-based, distance-based, and location-based algorithms. In each scheme, the decision of rebroadcasting depends on a predetermined threshold value. For instance, in the counter-based scheme a node will rebroadcast a message whenever the number of duplicate messages received before the node itself finds the medium free to transmit is below a certain threshold value. In this approach, a high number of duplicate messages received would indicate that other nodes in the neighborhood already rebroadcasted the message, so the current node must refrain from sending it to prevent the mentioned broadcast storm problem. The work presented in [Tseng *et al.* 2001] introduces a criteria to adaptively adjust the thresholds depending on the number of neighbors.

Oppositely to MANETs, there are just a few proposals of suppression techniques that aim specifically at VANETs. In particular, VANETs differ in many aspects from mobile ad hoc networks (MANETs). The movement of vehicles usually follows a common pattern, i.e., same or opposite directions. The mobility of vehicles is constrained to single or multiple-lane roads. These characteristics raise the need for new and specific solutions in vehicular environments.

In [Wisitpongphan *et al.* 2007], three broadcast suppression techniques are presented to alleviate the packet contention at the link layer. The proposed schemes work in a distributed manner without requiring information from other nodes in the neighborhood. They are illustrated in Figure 6.1 and will be briefly described below.

Weighted p-Persistence Broadcasting: whenever a new message is received from node i , node j will rebroadcast with probability p_{ij} defined as:

$$p_{ij} = \frac{D_{ij}}{R} \quad (6.1)$$

where D_{ij} is the relative distance between nodes i and j , and R is the average transmission range. In case the message has been received before, it is simply discarded.

Slotted 1-Persistence Broadcasting: whenever a new message is received from node i , node j will rebroadcast with probability 1 at the assigned time slot T_{Sij} defined as:

$$T_{Sij} = S_{ij} \times \tau \quad (6.2)$$

where τ is the estimated one-hop delay including the medium access delay and propagation delay, and S_{ij} is the assigned slot number expressed as:

$$S_{ij} = N_s \left(1 - \left\lceil \frac{\min(D_{ij}, R)}{R} \right\rceil \right) \quad (6.3)$$

where D_{ij} is the relative distance between nodes i and j , R is the average transmission range, and N_s is the predetermined number of slots. In case the message has been received before or a duplicate message is received before the node's assigned time slot, the message is simply discarded.

Slotted p-Persistence Broadcasting: whenever a new message is received from node i , node j will rebroadcast with probability p at the assigned time slot as expressed by Equation 6.2. Accordingly, in case the message has been received before or a duplicate message is received before the node's assigned time slot, the message is simply discarded.

Among the three schemes listed, the slotted 1-Persistence broadcasting achieved the best performance as described in [Wisitpongphan *et al.* 2007], reducing greatly the number of unnecessary broadcasts while still achieving low end-to-end delay and high reachability. As it has been described, all schemes described prioritize the rebroadcast of nodes with longer distances from the source in order to disseminate the message rapidly along the road.

It is important to emphasize that the suppression mechanisms mentioned are generally included in the network layer. Although they contribute greatly by mitigating the *broadcast storm problem*, collisions and delays will still occur because of the inherent probabilistic media access control (MAC) utilized by the *Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)* mechanism employed by IEEE 802.11 protocols [Schiller 2003], explained in Section 2.1. These mechanisms do not always guarantee that, for instance, a time slot assigned to a vehicle in the Slotted 1-Persistence scheme will be preserved. Specially in dense networks where the medium is likely to be found busy most of the time, the additional time randomness inserted by the backoff mechanism used in 802.11 MAC protocols may change the order determined by the time slots. Even though the delay introduced by the exponential back-off algorithm is much less (\ll) than the τ period defined for the suppression techniques above, the transmission of other nodes in the network (e.g., from other applications) may result in a long contention period for nodes waiting to transmit due to a busy medium and thus in the change of order of their transmissions. For instance, suppose nodes A and B are assigned to time slots 1 and 2, respectively. Node A will, therefore, immediately send a frame request to the MAC layer. If another node C is currently transmitting, node A will find the medium busy and run the back-off algorithm by waiting a DIFS period and an additional random number of time slots defined for the contention window. The number of time slots is only decremented when the medium is found idle for at least another DIFS period. At this time, B may have already sent a frame request to the MAC layer. If the medium is idle, it could happen that B's transmission is performed before A's transmission, because A may still be waiting for the number of time slots determined by the back-off algorithm. Another possibility is that another node D has been contending for access even before node A, and could consequently gain the access before A and introduces an even longer contention period to nodes A and B. Once again, the order of transmission between nodes A and B would be uncertain. In addition, in case the number of time slots randomly

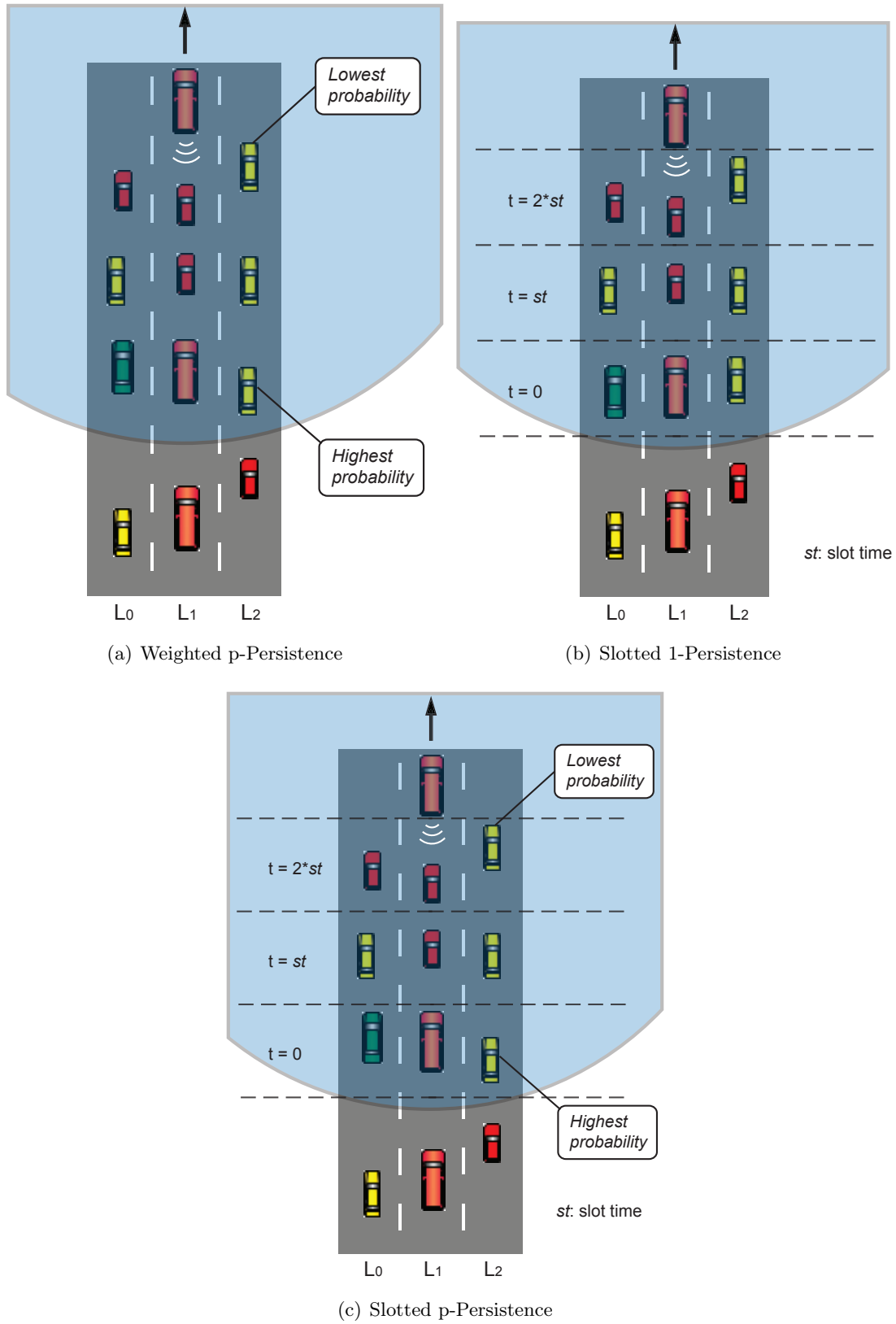


Figure 6.1: Illustration of the three broadcast suppression techniques proposed in [Wisitpongphan *et al.* 2007]

chosen in the backoff mechanism is the same for multiple nodes, collisions will still occur and affect negatively the spread of information in the network.

6.2 Addressing Source Vehicles

The broadcast suppression schemes outlined previously have been proposed as general solutions for vehicular communication. When addressing a specific application such as gathering information from vehicles to derive an overview of the traffic ahead, these solutions may need to be tailored. In fact, as explained in Chapter 5, two types of vehicles are considered by the OTHA protocol: *source* and *relay* vehicles. When a vehicular network with only relay vehicles is considered, any broadcast suppression technique would suffice, since vehicles are passing behind, i.e., relaying, the same¹ information over time. However, the introduction of *source* vehicles raises different aspects to be considered when designing our dissemination protocol. Differently from relay vehicles, source vehicles contain crucial information to be included in the TrafficMap being exchanged and must be always considered. Hence, in this context our dissemination protocol must behave adequately according to the role of vehicles and not only, for instance, prioritize vehicles positioned further away in order to broadcast the information as quick as possible.

In this work, we base our design for the dissemination protocol layer on the protocol solution for a single-lane road described in [van Eenennaam 2008]. The suppression strategy is mainly based on the *slotted 1-Persistence* suppression technique proposed in [Wisitpongphan *et al.* 2007]. In order to cope with *source* vehicles, early time slots are allocated for them. Because they possess critical information, these early time slots give them the opportunity to transmit quickly and cancel TrafficMap Messages scheduled by relay vehicles.

Another important characteristic one must consider is the speed dependence among vehicles. A reduction in speed by vehicles ahead on the road may induce a reduction in speed by vehicles behind going to that direction as well. Therefore, it is reasonable to assume the existence of a logical order of events that start with vehicles ahead on the road upstream to vehicles behind that are about to meet these events. Due to this inter-speed dependence, when considering time slots reserved to source vehicles, oppositely to having the most distant source vehicles from the sender rebroadcasting first, source vehicles closer to the sender will be preferred. This is reasoned by the fact that source vehicles are the vehicles responsible for detecting and reporting such events which in our case are speed deviation on the road. In fact, within the transmission range employed there might be several vehicles about to detect a speed deviation by means of the *sensitivity* ε function. Because of the mentioned speed dependence among vehicles, it is possible that the TrafficMap Message sent by the closest vehicle to the sender changes the role of vehicles behind to behave now as relay vehicles. Relay vehicles which are assigned to later time slots, on the other hand, will behave as defined by the *slotted 1-Persistence*, as the objective is to spread the information as quick as possible.

Due to the importance given to source vehicles, we do not allow their broadcasts to be suppressed, i.e., canceled upon the receipt of another broadcast. This is guaranteed by relying on unique IDs assigned to source vehicles as we describe in detail in Section 6.3.2. One interesting

¹Although relay vehicles can still contribute by averaging the last entry added to their lane in the TrafficMap (see Section 5.3.3), such improvement is not crucial for the correct representation of the road ahead.

consequence of this design decision is that since messages will be broadcasted asynchronously by different source vehicles, a flow initiated by the head vehicle of a cluster might be split into multiple TrafficMap information flows along the road. On the one hand, multiple TrafficMap information flows results in a less time efficient protocol because a higher number of messages might be introduced into the network. On the other hand, this measure clearly prioritizes accuracy, since every speed deviation detected by a vehicle is broadcasted. A means to merge and diminish the number of simultaneous information flows on the road is described in Section 6.3.1.1.

In [van Eenennaam 2008] the utilization of a higher priority given to source vehicles has resulted in longer delays without much gain in accuracy. However, a single ID has been utilized for each flow initiated despite the existence of source vehicles. The result is that within a transmission range only one source vehicle would be considered, since the transmission of all other vehicles would be suppressed after hearing a message with the same ID they are about to use. This is not sufficient to capture every speed deviation in situations where there are multiple source vehicles with different speed deviations to be reported, for instance, in a simple scenario with multiple lanes. Differently, our strategy guarantees that every speed deviation is reported by means of the unique ID used by source vehicles. In addition, the priority given to source vehicles in our work is supported by the mentioned influence of events that occur from downstream to upstream, i.e., a TrafficMap Message sent by the closest source vehicle with regard to the sender in the suppression mechanism may change the role of vehicles behind.

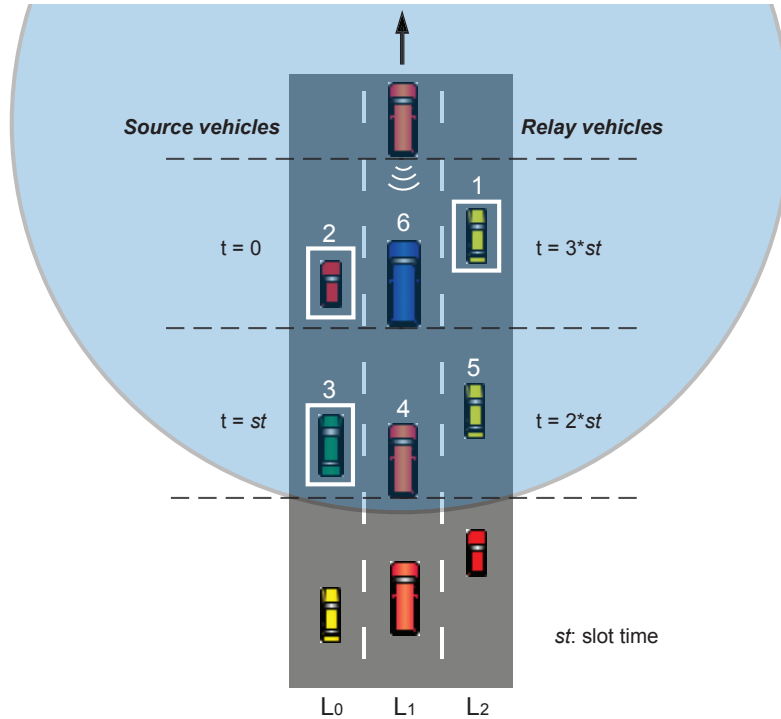


Figure 6.2: Overview of the dissemination protocol proposed

The overview of our suppression strategy functioning is depicted in Figure 6.2. In this example, four time slots are utilized: the two earliest reserved for source vehicles while the remaining reserved later for relay vehicles. Source vehicles are marked with a rectangle surrounding each of them. The number above each vehicle indicates their turn according to their assigned time slot. As indicated, the broadcast performed by relay vehicles functions the same as the *slotted 1-Persistence* protocol. The most distant relay vehicles are assigned to the first time slot reserved for relay vehicles: $t = 2 \times st$. Differently, source vehicles have the opposite pattern with the closest vehicles to the sender having the earliest time slot assigned to them: $t = 0$.

6.3 The Dissemination Protocol Layer

After motivating and adapting the *slotted 1-Persistence* suppression technique to work properly with vehicles acting either as *source* or *relay* vehicles, we describe in this section the Dissemination Protocol Layer.

Figure 6.3 depicts the functioning of the protocol layer. A message is first received from the lower layer which is envisioned to be the MAC Layer defined by IEEE 802.11p, the upcoming IEEE standard for vehicular communication (see Section 2.2). The first decision process verifies whether the message has been originated by a vehicle further in the message direction. The message direction is included in each TrafficMap Message and defined by the application running on top of the OTHA protocol, i.e., the Congestion Assistant. For this moment we consider the direction of every message being towards vehicles further behind. In addition, in this protocol we consider that vehicles only participate in rebroadcast with regard to their own message direction, which is assumed to be opposite to their driving direction. If the message has been originated by a vehicle further in message direction, the vehicle processing the message will verify whether the message is a rebroadcast of a previous broadcast b . In case it in fact refers to a previous rebroadcast, if the receiver vehicle has scheduled a message for the same broadcast b and it is simply relaying information, such message will be canceled. This means some other vehicle behind already had the opportunity to spread the TrafficMap data and no more messages are necessary. The protocol does not cancel messages of vehicles scheduled to behave as source vehicles, since they contain critical information to be transmitted as explained previously. Oppositely, when a message is originated from a vehicle not further in the message direction, i.e., it comes in fact from some vehicle ahead on the road, a decision process evaluates whether that message has already been seen before. If it is an old message, it is simply discarded. Otherwise, the message is passed to the upper Traffic Filter Protocol Layer to be further analyzed.

After the message is processed by the Traffic Filter Protocol Layer, there may be a message request sent back to the Dissemination Protocol. When a message request arrives, it is first handled by the *Time Manager*. This process is responsible for controlling the sending of messages into the network. It in fact defines a lower and upper bound for the time a message must wait before being broadcasted. It is also meant to assign the time slot of our suppression technique to each vehicle. The last step of the protocol is the preparation of the message to be sent down to the MAC layer by the *Message Builder*. This process is responsible for defining the message header and acquiring the latest TrafficMap data available in the vehicle's memory managed by the Traffic Filter Protocol Layer.

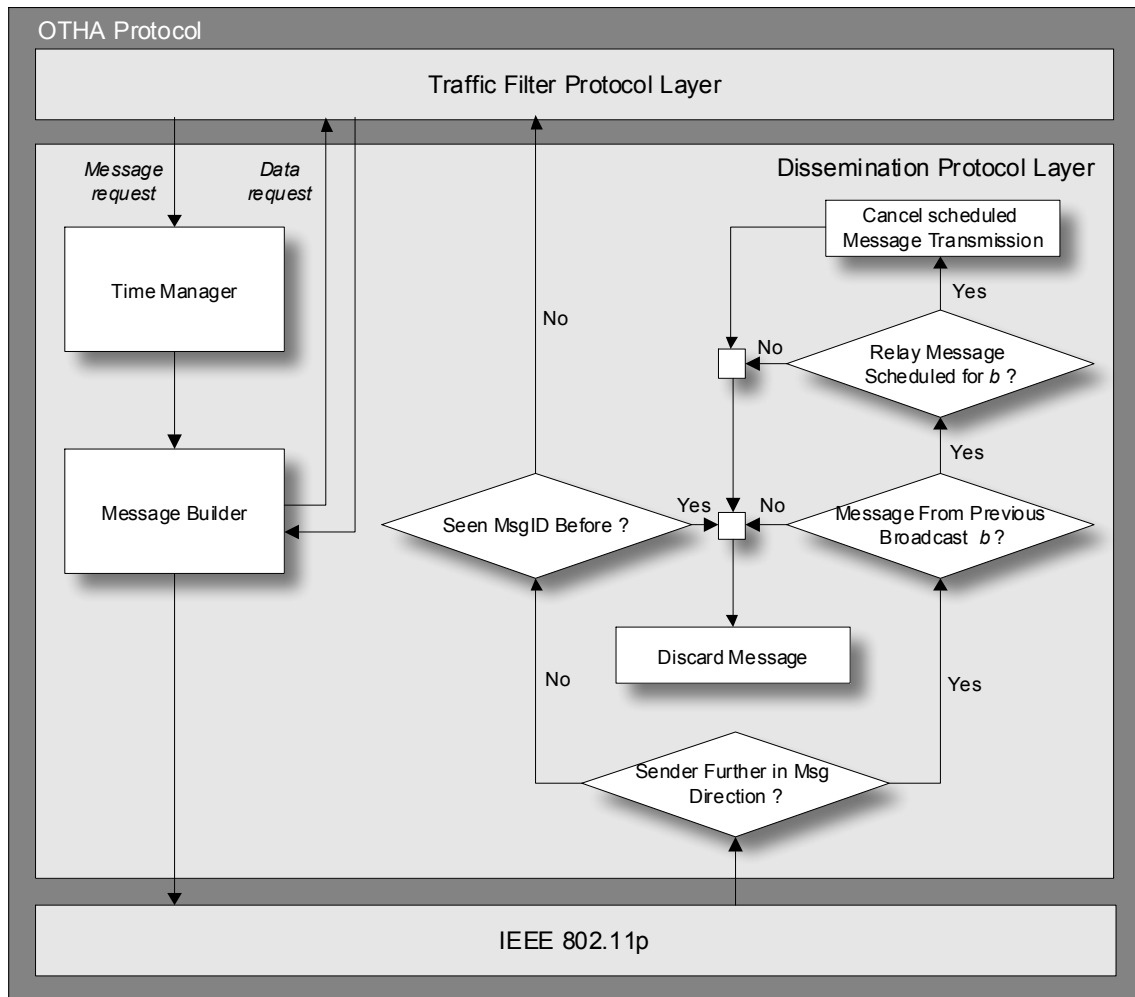


Figure 6.3: The Dissemination Protocol Layer

In the sequel, we complete the description of the Dissemination Protocol Layer by detailing the *Time Manager* and *Message Builder* processes.

6.3.1 Time Manager

Timing is of great importance when delivering up-to-date information to every vehicle is a prime concern. The OTHA protocol must provide means for the Congestion Assistant application running on top of it to be able to set the required maximum period a vehicle should wait before receiving new information about the traffic ahead. At the same time, our protocol must also leave some bandwidth for other more time critical applications.

In order to meet both requirements, we rely on two timers in our Dissemination Protocol Layer which have been proposed in [van Eenennaam 2008]: the τ timer and the *FFP timer*. In addition, we define the *Broadcast Suppression Timer* which is set to the time slot duration assigned to a vehicle. In the following Sections 6.3.1.1 and 6.3.1.2, the timers employed in our protocol as well as the *Time Manager* process are explained, respectively.

6.3.1.1 The Timers

τ timer

It establishes a maximum inter TrafficMap Message (MIT) period. It ensures an upper-bound for the time vehicles have to wait before receiving a message, i.e., it guarantees the periodicity determined by the Congestion Assistant. Whenever a vehicle does not receive any relevant information (where the relevancy is determined by the Traffic Filter Layer) before the τ timer expires, a new message is broadcasted. Such timer will mostly expire in vehicles driving at the beginning of a vehicle cluster, since no new information from the traffic ahead will be received until another vehicle ahead on the road is encountered. Vehicles positioned within a cluster will periodically receive a message from other vehicles down the road and reset their τ timers. Due to this fact, vehicles which have their τ timers expired are the ones classified as *flow initiator* vehicles.

Even though the MIT period may be defined by the Congestion Assistant, this period must be upper bounded by a value short enough for the system to be able to react to sudden changes in traffic dynamics. Giving that the Active Pedal function of the Congestion Assistant operates at a distance of 500-1500m from the tail of the traffic jam, any sudden change on the traffic ahead must be reported to vehicles behind in a such period of time they are able to react properly, for instance, by braking before reaching the tail of the jam. A simple worst case scenario described in [van Eenennaam 2008] that considers the so-called wide moving jams [Kerner & Rehborn 1996] estimates that a TrafficMap Message must be received and reported to the Congestion Assistant at latest at every 3.0 seconds for a proper reaction to sudden changes, e.g., crashed vehicles down the road. In this work, this estimate will also be assumed and used as the default MIT period.

FFP timer

It stands for *Flood Free Period*. It ensures a lower-bound for the time between two consecutive transmissions of a vehicle. After each transmission, the FFP timer is activated and only after its expiration the vehicle is able to send another message, if it is required. Such timer helps limiting the existing total number of messages in the network per time unit, i.e., it establishes the maximum transmission frequency, what gives other applications the opportunity to use the radio channel. A proper value for the interval defined for the timer requires a more thorough evaluation of the OTHA protocol running with other applications simultaneously and it is left for future work. An interesting aspect of this timer is that the *merging* process described in Section 5.3.1 benefits from the additional waiting time between two consecutive transmissions. Due to the likely existence of multiple information flows being propagated along the road, either near a junction point or simply within a multiple-lane road, these flows are given the chance to be merged into a single one whenever a vehicle is currently waiting for the FFP timer to expire. This results in a more bandwidth efficient protocol, since fewer information flows means less frequent transmissions per vehicle.

Broadcast Suppression Timer

The remaining timer defined for our protocol regards the time slot assignment of our broadcast suppression technique summarized in Section 6.2. Upon the receipt of a message, if the Traffic Filter Protocol Layer requests a new rebroadcast and the FFP timer is not set, the *Broadcast Suppression Timer* will define the time the vehicle needs to wait before sending its message down to the MAC layer.

As motivated previously, the total number of time slots are divided into time slots reserved for source vehicles and time slots reserved for relay vehicles, where the earliest time slots are given to source vehicles due to their higher priority. Based on the *slotted 1-Persistence* suppression technique proposed in [Wisitpongphan *et al.* 2007], the assignment of time slots is determined by the distance between the receiver vehicle which is about to rebroadcast and the sender of the message it has just been received. Among source vehicles, the closer a vehicle is from the sender the earlier will be its time slot. The time slot distribution for relay vehicles functions in the opposite pattern, with the farthest vehicles from the sender obtaining the earliest time slots.

The time slot assignment is defined as follows. A vehicle j when receiving a message from vehicle i first calculate the percentage distance PD_{ij} between both vehicles with respect to the estimated transmission range R .

$$PD_{ij} = \left\lceil \frac{\min(D_{ij}, R)}{R} \right\rceil \quad (6.4)$$

where D_{ij} is the relative distance between nodes i and j . As a result, the PD_{ij} value will vary within the interval $[0,1]$ with large distances being closer to 1.

Because of the different time slot assignment for source and relay vehicles, we define separate formulas for each of them. The time slot number assigned to source vehicles $S_{source_{ij}}$ is defined as follows:

$$S_{source_{ij}} = \begin{cases} 0 & \text{if } PD_{ij} = 0 \\ \lceil NS_{source} \times PD_{ij} \rceil - 1 & \text{if } PD_{ij} > 0 \end{cases} \quad (6.5)$$

where NS_{source} is the total number of time slots reserved for source vehicles. If vehicles are uniformly distributed within the transmission range of vehicle i , they will be equally distributed among the NS_{source} time slots reserved. $S_{source_{ij}}$ will vary within the interval $[0, NS_{source} - 1]$.

The time slot number assigned to relay vehicles $S_{relay_{ij}}$ is defined by the following equation:

$$S_{relay_{ij}} = \begin{cases} NS_{source} & \text{if } PD_{ij} = 1 \\ NS_{source} + (\lceil NS_{relay} \times (1 - PD_{ij}) \rceil - 1) & \text{if } PD_{ij} < 1 \end{cases} \quad (6.6)$$

where NS_{relay} is the total number of time slots reserved for relay vehicles. Note that $S_{relay_{ij}}$ starts from time slot number NS_{source} , as relay vehicles will always transmit after source vehicles. Hence, $S_{relay_{ij}}$ will vary within the interval $[NS_{source}, NS_{source} + (NS_{relay} - 1)]$.

Based on the time slot numbers $S_{source_{ij}}$ and $S_{relay_{ij}}$, the total time source and relay vehicles have to wait before rebroadcasting is given by equations 6.7 and 6.8, respectively.

$$T_{source_{ij}} = S_{source_{ij}} \times st \quad (6.7)$$

$$T_{relay_{ij}} = S_{relay_{ij}} \times st \quad (6.8)$$

where the slot time st is an over-estimation of the one-hop delay including the medium access delay and propagation delay.

The assignment of different time slots to vehicles depending of their positions clearly breaks the synchronization present in the simple flooding approach, where all nodes would rebroadcast simultaneously upon the receipt of a message. The slot time st is defined in a manner it gives vehicles assigned to later time slots the opportunity to cancel their transmissions, since the message has already been rebroadcasted. Therefore, ideally only vehicles of a single time slot among the total number of time slots would rebroadcast. However, a similar synchronization on a smaller scale can still occur within this single time slot when multiple vehicles are assigned to the same time slot and will all start their transmission simultaneously.

Such synchronization within a time slot has been identified by van Eenennaam in [van Eenennaam 2008]. In order to cope with it, he proposes a variation of the slotted 1-Persistence broadcasting scheme that has been referred to as *microSlotted 1-Persistence Flooding*. The proposed scheme functions the same as the Slotted 1-Persistence Broadcasting scheme but with a small additional delay, i.e., the micro slots, within each time slot to break the mentioned synchronization.

The same problem has been identified and referred to as the *Timeslot Boundary Synchronization Problem* in [Blum & Eskandarian 2009]. This work describes design guidelines for both link and network layers in order to avoid such synchronization problem. It is argued that, besides the fact that near simultaneous receipt and rebroadcast of a message can cause synchronization, network congestion contributes greatly to synchronization as well. If one transmission causes

multiple nodes to freeze their back-off timers in the MAC layer, the timeslot boundaries for these nodes will be synchronized when they detect that the medium has become idle. The guidelines for both communication layers rely on an additional delay before each transmission. For the MAC layer, an additional pseudo-random delay to SIFS, which is the delay following a transmission that a node must wait before restarting its back-off timer, is proposed. For the network layer, the delay may be a function of the distance, as we define for our Dissemination Protocol Layer, however, it should be chosen from a near continuous interval in order to break completely the alignment of timeslot boundaries.

In our work, we limit ourselves and concentrate our efforts in avoiding the synchronization problem only in the OTHA protocol, i.e., above the MAC layer. Instead of relying on time slots on a small scale, as proposed in [van Eenennaam 2008] with the introduction of micro slots, we simply define a small additional delay still as a function of the distance but from a near continuous interval as suggested in [Blum & Eskandarian 2009]. Note that such extra delay inserted within a slot time is meant only to mitigate the mentioned synchronization problem and does not contribute with the suppression of other broadcasts in the network, as it is defined as only a very small fraction of a single slot time.

The additional delay is defined for both source and relay vehicles by equations 6.9 and 6.10, respectively. For source vehicles, the additional delay $AD_{source_{ij}}$ is defined as follows:

$$AD_{source_{ij}} = D_{max} \times PD_{ij} \quad (6.9)$$

where D_{max} is the maximum allowed delay. D_{max} must be strictly smaller than st to avoid that the time slots assigned to vehicles overlap each other and at the same time it must still give time to vehicles assigned to later time slots to cancel their transmissions. This is possible, since we choose an over-estimation for the one-hop delay st as described previously. In fact, in this overestimation both the transmission time (maximum delay introduce by the MAC Layer) and the defined D_{max} are considered.

The additional delay for relay vehicles $AD_{relay_{ij}}$ works similarly but with farther vehicles obtaining a smaller delay as defined in equation 6.6 for their time slot number.

$$AD_{relay_{ij}} = D_{max} \times (1 - PD_{ij}) \quad (6.10)$$

The total time source and relay vehicles have to wait before rebroadcasting is updated to include the additional delay described as follows in equations 6.11 and 6.12.

$$T_{source_{Sij}} = (S_{source_{ij}} \times st) + AD_{source_{ij}} \quad (6.11)$$

$$T_{relay_{Sij}} = (S_{relay_{ij}} \times st) + AD_{relay_{ij}} \quad (6.12)$$

An illustration of the functioning of our time slot assignment is given in Figure 6.4. In this example, we define two time slots for source vehicles and three time slots for relay vehicles. Figure 6.4(a) shows how the time slot number is distributed among vehicles as a function of their percentage distance from the sender. The earliest time slots 0 and 1 are assigned to source vehicle while the remaining 2,3 and 4 are reserved to relay vehicles. We can notice that for each

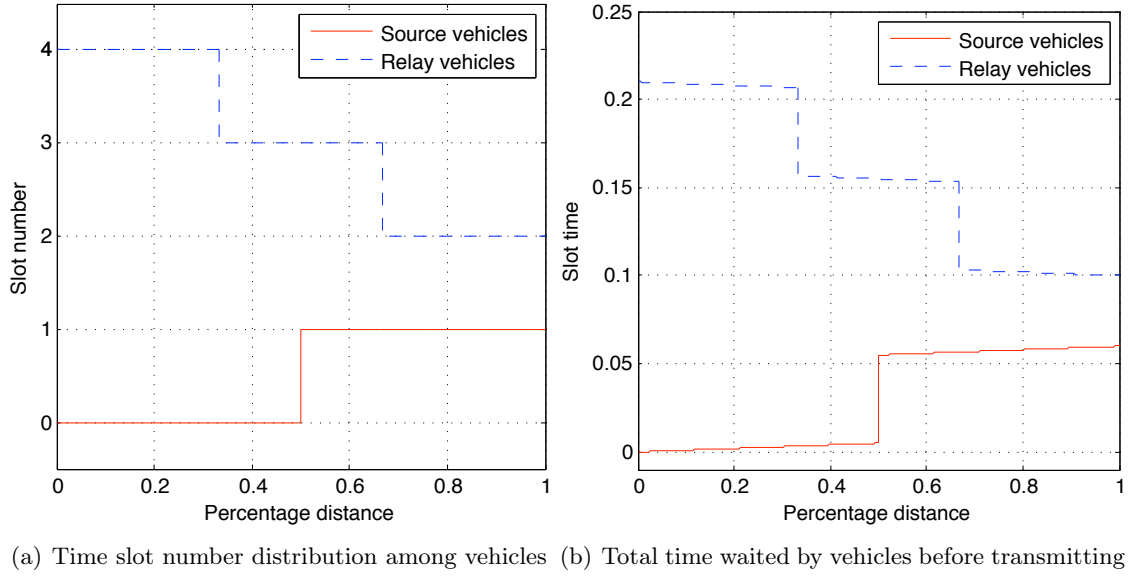


Figure 6.4: Illustration of the time slot assignment with two slots for source vehicles and three for relay vehicles

type, the vehicles are equally distributed within the total transmission range. The difference in how each type of vehicle has its time slot assigned is also evident, with source vehicles closer to the sender receiving earlier time slots and relay vehicles having the opposite pattern. Figure 6.4(b) shows the distribution of the total time assigned to each type of vehicle. Note that the additional delay proposed by the equations 6.11 and 6.12 provides a continuous increase in the delay as a function of the percentage distance, which gives vehicles different slot times, unless they are exactly at the same distance from the sender.

Relation between the timers

The choice of proper values for the timers described depends not only the requirement specifications of the application running on top of the OTHA protocol, i.e., the Congestion Assistant, which is the case for the τ timer, but also on further analysis on the performance of the protocol in different road scenarios, as it is required in order to define the FFP timer value, for instance. We leave such further analysis on the proper definition of each timer value as future work.

Nevertheless, there is a relation between these values that must be followed in order to have the protocol running properly. The broadcast suppression timer and the FFP timer must be defined as a function of the τ timer, which although upper bounded (see explanation of the τ timer) is independently defined by the application. The FFP timer value must be rather small, for instance, strictly smaller than the τ timer value, since great values would probably impede vehicles far upstream to receive up-to-date information. This may happen due to the split of TrafficMap flows along the road, i.e., flows arriving later may be delayed by the FFP

timers set by vehicles during previous flows. On the other hand, great values for the FFP timer would increase the probability of merging these independent flows along the road, and thus the bandwidth utilized by the OTHA protocol would be lower. The FFP timer value must also assume values greater than the maximum time defined by the latest time slot in the equations 6.11 and 6.12, since otherwise its value would not limit effectively the transmission rate as it is meant to. We establish then the following basic relation between the timers:

$$t_{broadcast} < t_{ffp} \ll t_{\tau} \quad (6.13)$$

where $t_{broadcast}$, t_{ffp} and t_{τ} are determined as the maximum possible value assigned by the broadcast suppression timer, the FFP timer value and the τ timer value, respectively. We define the t_{ffp} as being *much less than* t_{τ} in order to prevent long multi-hop delays. In addition, if a flow originated by a *flow initiator* vehicle is split into multiple flows along the road, these flows would be separated at some point by only a small fraction of time, determined by time differences in the transmission delay during the broadcast suppression algorithm. Therefore, t_{ffp} could be defined as a function of $t_{broadcast}$, in order to compensate for the transmission periods that separate these different flows. An interesting relation between the FFP and τ timers has been derived in [van Eenennaam 2008]. The relation is that, as one may notice from the above equation, t_{ffp} in fact limits the number of transmissions per vehicle during a MIT period (defined by t_{τ}) by the upper bound NT_{max} defined as:

$$NT_{max} = \frac{t_{\tau}}{t_{ffp}} \quad (6.14)$$

Another aspect is that, it may occur that different flows initiated by a common *flow initiator vehicle* meet along the road, i.e., the head of a new flow with the tail of a previous. This may happen due to the different end-to-end delays induced by $t_{broadcast}$ and t_{ffp} . For instance, even if only the $t_{broadcast}$ is considered, it may occur that most vehicles are assigned to the latest time slot during a first flow, whereas in a subsequent flow most vehicles are assigned to the earliest time slot, e.g., because a constant change in traffic dynamics. The sum of delays introduced during the former flow may be large enough to shorten the difference between the two flows and provoke a meeting between them. This could be aggravated by t_{ffp} as it is greater than $t_{broadcast}$. Nevertheless, since t_{τ} is set to be much greater than both mentioned values, the likelihood of this meeting may be considered small as the existence of many hops would be required for such large delay to be introduced. In order to further understand the actual probability of these events further study is required. Despite this undesirable possibility, the major consequence of such meeting would be that part of the vehicles would only receive the later flow (in the example described). Despite the longer end-to-end delay introduced, they would still receive up-to-date information. Another minor consequence is that since these vehicles would have waited a period longer than t_{τ} to receive new information, they would start a new flow themselves unnecessarily. However, this may be normalized when new flows from down the road arrive.

6.3.1.2 The Time Manager Process

After describing each timer utilized by our Dissemination Protocol, we are able to explain the *Time Manager* process, as depicted in Figure 6.5. After a message request is received from the Traffic Filter Protocol Layer, a decision is processed to check whether the FFP timer is set or not. In case it is set, the message is held back until the FFP timer expires. Otherwise, if there is not a broadcast already scheduled, the appropriate time slot is defined by following equations 6.11 or 6.12 depending on the vehicle's type, and scheduled accordingly. When the FFP timer expires and there is a message currently requested, the τ and FFP timers are reset. As described in this process, the FFP timer is only reset when there is a message to be transmitted. Since we are interested in establishing a lower bound for the time between two consecutive transmissions, if a message is not requested and the FFP timer expires there is no need for resetting it. In case the τ or the broadcast timers expire, the FFP timer and the τ timer are reset. The τ timer is reset at this point of the protocol, when there is a message to be broadcasted and not directly when a message is received, because the upper layer can still discard received messages if it does not contain any relevant information, e.g., when it receives a message from the opposite direction. When a message request is sent to the Dissemination Protocol Layer there is a guarantee that either the vehicle is participating in a rebroadcast or the vehicle is currently a *flow initiator* and had its τ timer expired.

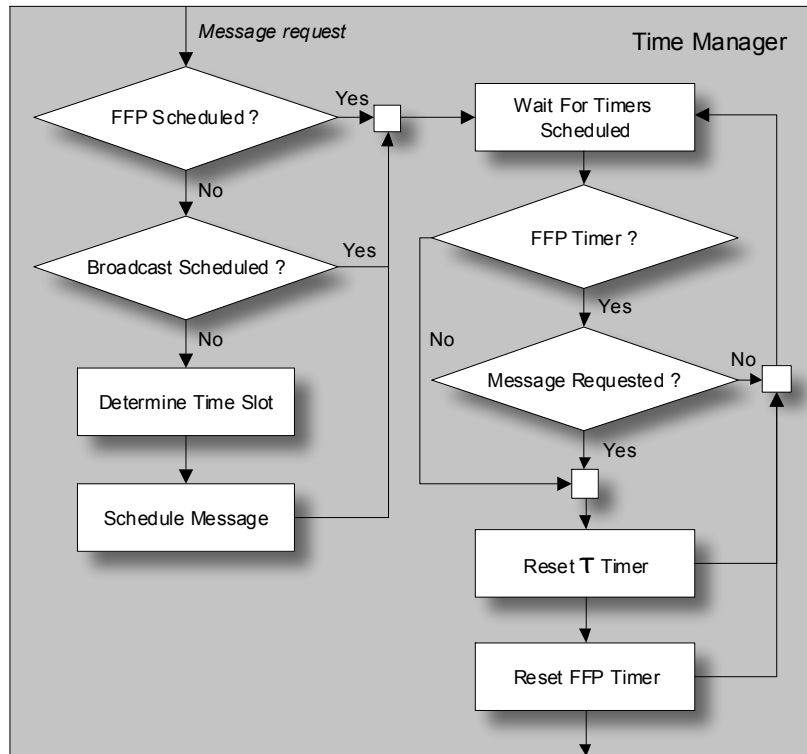


Figure 6.5: The Time Manager

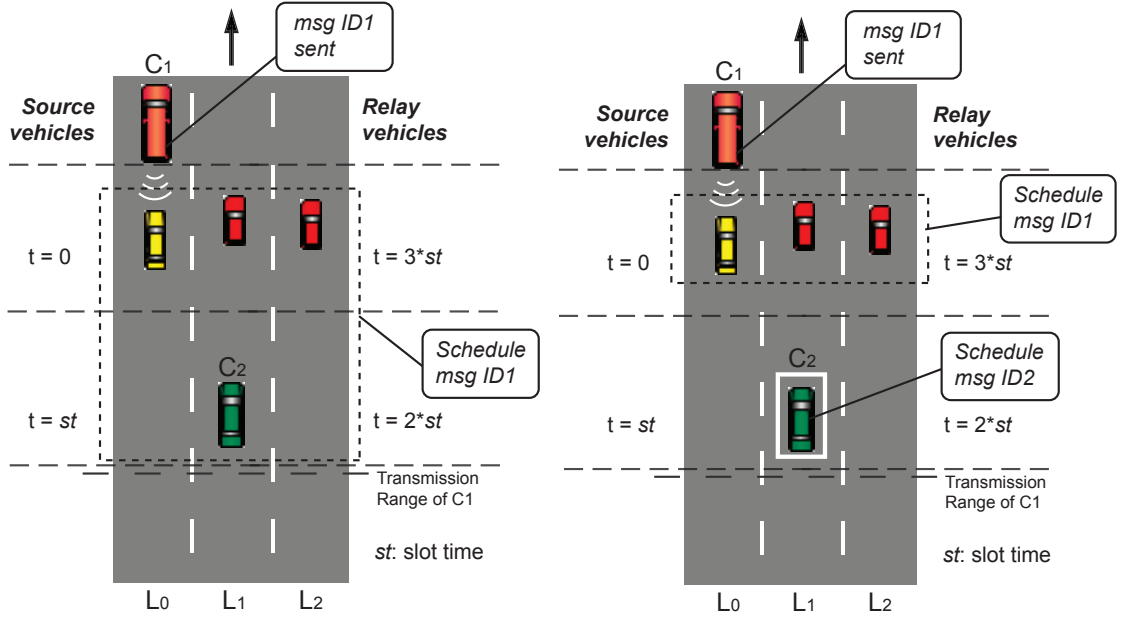
An important remark regarding the τ timer is that it must be defined in a way it does not unnecessarily expire in vehicles within a cluster. Ideally, only vehicles at the beginning of a cluster, the *flow initiator* vehicles, should broadcast a new message due to a τ timer expiration. In fact, if every vehicle set its τ timer to expire after the same amount of time, vehicles positioned within a cluster could have their timer expired very often simply because of small end-to-end delay variations from the *flow initiator* up to them. For instance, let us assume that the current *flow initiator* C_{FI} broadcasts a message at time instance $t = x_0$ and that a vehicle C_1 within the existing cluster receives this first message at $t = x_0 + d_1$, where d_1 is the end-to-end delay from C_{FI} to C_1 measured at this time instance. Supposing that both vehicles reset their τ timers to expire after a defined Maximum Inter TrafficMap Message period (MIT) of 3.0 seconds, i.e., at $t = x_0 + 3.0$ and $t = x_0 + d_1 + 3.0$, for C_{FI} and C_1 , respectively, any delay $d_2 > d_1$ in the next end-to-end propagation between both vehicles would result in C_1 having its τ timer expired. In fact, due to the inherent random back-off timer employed by the underlying 802.11 MAC protocol layer, the end-to-end path delay between vehicles in the network may change frequently. In order to avoid that vehicles within a cluster have their τ timers expired, we define they will set their τ timers to expire later than *flow initiator* vehicles. This can be done by including an additional random fraction of the pre-defined MIT value to their expiration time.

6.3.2 Message Builder

The correct identification (ID) of TrafficMap Messages is crucial for the Dissemination Protocol to work properly. While the suppression techniques described in Section 6.1 rely on the existence of a single message being spread along the road and thus a single message ID, every message broadcasted by source vehicles contains important information that all vehicles upstream, without exception, must receive. Hence, messages transmitted by source vehicles are given a unique ID in order to differentiate them from simply relayed messages. The result is that our protocol is in fact two-fold: (i) source vehicles initiate new broadcasts by sending out messages uniquely identified; (ii) relay vehicles reuse previous message IDs assigned by source vehicles and participate in a rebroadcast process.

The unique identification of messages sent by source vehicles brings new challenges with respect to the canceling of messages. Figure 6.6 compares a scenario with only relay vehicles participating in the rebroadcast of a message sent by vehicle C_1 with a situation where one source vehicle is present and it starts a new broadcast by sending a message with a unique ID. In the former situation (a), vehicle C_2 which is positioned further in message direction would have the chance to rebroadcast before the remaining vehicles which in turn could cancel their transmissions, since vehicle C_2 already rebroadcasted. On the other hand, in the latter scenario (b), the source vehicle C_2 would also have the chance to broadcast first, however, now the remaining relay vehicles would receive a message with a different ID from what they have seen before and would not be able to cancel their transmissions.

It is evident that the unique message IDs used by source vehicles must be addressed in a way the rebroadcast of relay vehicles are still canceled. Otherwise, redundancy would be unnecessarily increased, which is not acceptable. One straight-forward manner of dealing with the canceling of messages is by simply discarding messages scheduled by relay vehicles whenever they realize some other vehicle already broadcasted some message behind them. In this way, instead



(a) Broadcast with only relay vehicles receiving (b) Broadcast with source and relay vehicles receiving

Figure 6.6: The unique source ID problem

of comparing the ID they are about to use with the one heard from some vehicle upstream, they would simply discard their transmission upon the receipt of any message further in the message direction. Of course, if both scheduled and received messages are meant to be propagated to the same direction. Although not obvious, there are in fact cases where this approach does not work correctly and yet compromises the reachability of the spread information. One of such cases is shown in Figure 6.7. In this example, in (a) vehicle C_1 transmits and vehicles C_2 and C_3 , which are within the transmission range of C_1 , schedule a broadcast as source vehicles by setting unique message IDs. Assuming both transmissions are scheduled for the same time slot but with a small delay difference, one transmission cannot cancel the other, since both messages have already been sent down to the MAC Layer. Supposing the medium has been busy and the MAC layer chooses a random smaller backoff time for vehicle C_3 (a possibility raised in Section 6.1), in (b) the message is received by vehicle C_4 , which in turn schedules a message as a source vehicle. Immediately after the medium is found free by vehicle C_2 , in (c) it broadcasts and vehicle C_3 schedules a new message now as a relay vehicle in order to propagate C_2 's message further in message direction. At this point, vehicle C_4 does not receive such message because it is out of the transmission range of C_2 . Finally, in (d) vehicle C_4 has its time slot finished and message sent. Vehicle C_3 , which has lately scheduled a message as a relay vehicle, cancels it since it heard some other vehicle (C_4) already broadcasting a message further in message direction. Clearly, this situation is not desirable, as C_3 was in fact holding new information with respect to vehicle C_2 and could not propagate it further.

The solution we propose is simply an extension of the canceling approach described for the mentioned suppression techniques in [Wisitpongphan *et al.* 2007]. Normally, all vehicles

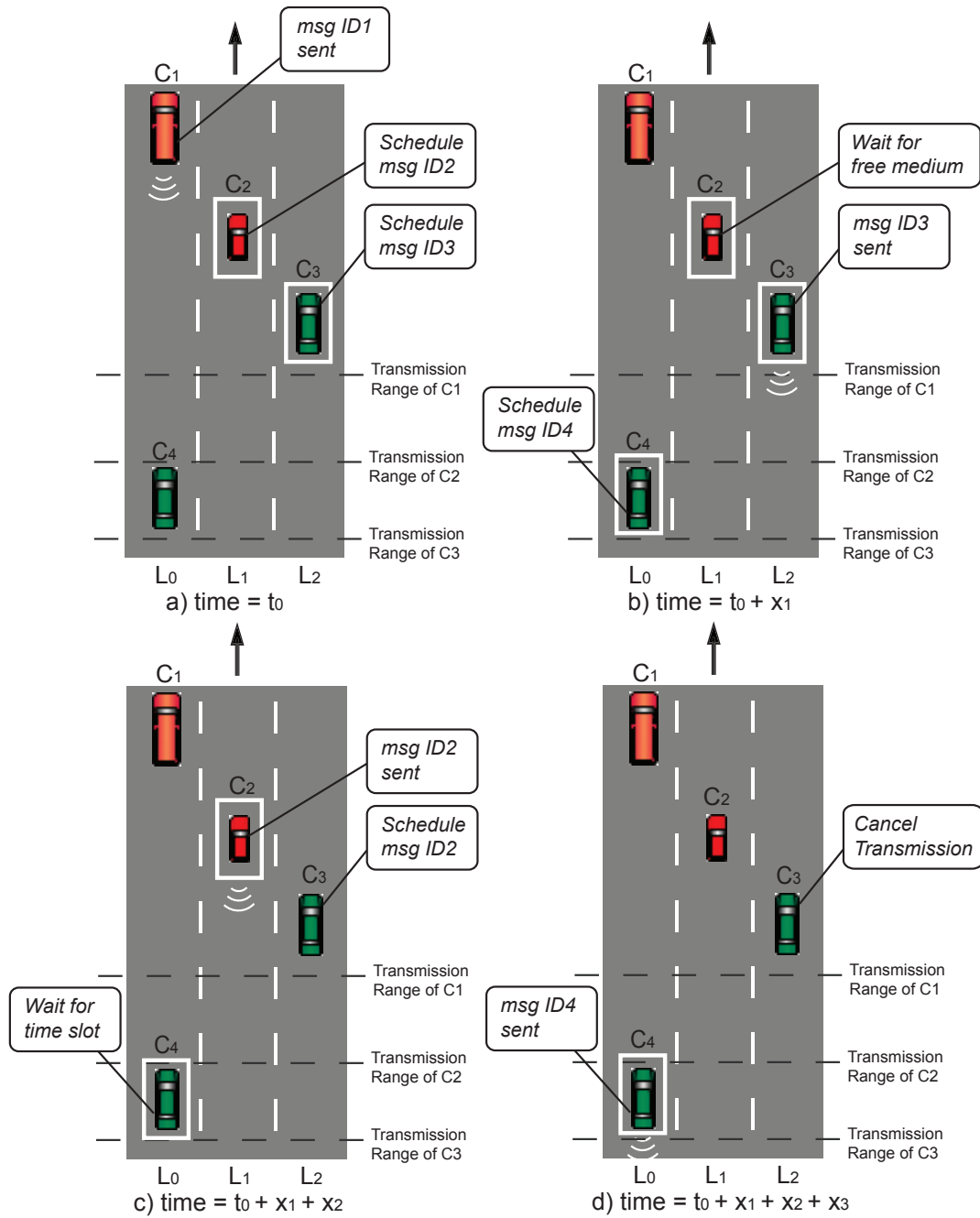


Figure 6.7: The problem of simply canceling messages just based on the type of vehicles. The information contained in *msg ID2* is lost because vehicle C_3 cancels its transmission after hearing *msg ID4*.

which receive a broadcast message schedule a rebroadcast for that message and cancel their transmission as soon as they hear some vehicle already rebroadcasting with the same message ID. In order to include source vehicles in this solution, we define two message IDs for each message: the *msgID* and *msgID_prev* identifications. When a vehicle acts as a relay vehicle, it simply repeats the last *msgID* received into both *msgID* and *msgID_prev* fields. On the other hand, source vehicles include the new ID into the *msgID* and repeat the last *msgID* received into the *msgID_prev* field. A vehicle can only cancel its own transmission if the message receive further in message direction contains the same *msgID_prev* as the *msgID* the vehicle is about to use. As a result, we still give source vehicles a unique ID and at the same time provide means for relay vehicles to cancel their messages only when other vehicles behind already rebroadcasted the information they hold.

The solution proposed is illustrated in Figure 6.8. In (a), for the same situation depicted previously in Figure 6.6(b), when vehicle C_2 , participating in the rebroadcast started by vehicle C_1 , broadcasts its message, all the remaining vehicles are able to identify by means of the *msgID_prev* field that the message received refers to the same message they have scheduled as an answer to C_1 's broadcast. They can then safely cancel their transmission. In (b), the same occurs for the latter situation presented in Figure 6.7(d). Instead of having vehicle C_3 canceling its message scheduled with the content previously added by C_2 , since the *msgID_prev* = ID3 received from C_4 is different from C_3 's current *msgID* = ID2, no messages are canceled and the information originated from C_2 remains scheduled.

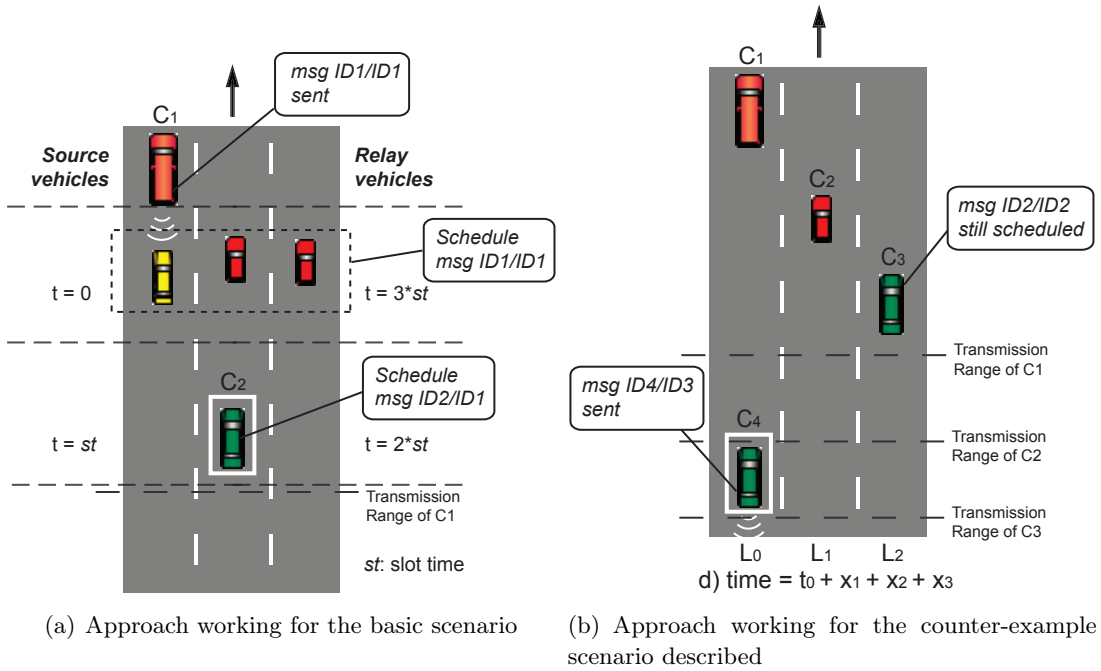


Figure 6.8: A solution for the source ID problem

The complete message structure is presented in Figure 6.9. The message header comprises the mentioned message IDs *msgID* and *msgID_prev*. Each message ID is formed by the fields:

vehicleID and *seqNumber*; and *vehicleID_prev* and *seqNumber_prev*, respectively. The *vehicleID* field value could be the MAC address of the vehicle or some other kind of unique identification. The sequence number *seqNumber* guarantees a unique identification for each message sent by an individual vehicle. The remaining fields are the *senderCoordinate* and *msgDirection* which are used to verify whether the vehicle analyzing the message is further or not in the message direction. The *msgDirection* value could be represented by a coordinate further in message direction on the road where the message should travel to. The payload of the message basically includes the TrafficMap Structure explained in Chapter 5.

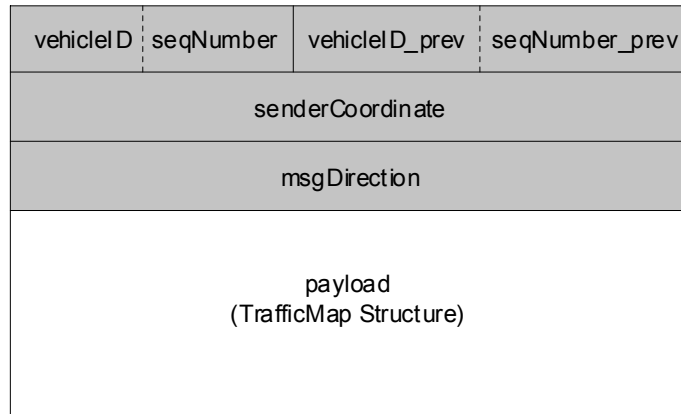


Figure 6.9: The message structure

Based on the observations given regarding the reasoning behind the message structure and its values, we describe now the *Message Builder*. The Message Builder is in fact a simple process that constructs the message to be sent based on the role of vehicles, i.e., whether they will act as source or relay vehicles, as illustrated in Figure 6.10. If it is a source vehicle, a unique value for the message ID is prepared: first by adding the vehicle's own ID followed by a new sequence number. Otherwise, relay vehicles simply repeat the last vehicle ID received into the message's vehicleID field. In the following, for both type of vehicles the last msgID value received is copied into the *msgID_prev* field, i.e., *vehicleID_prev* and *seqNumber_prev* values. In order to fill in the payload, the latest TrafficMap content is retrieved from the Traffic Filter Protocol Layer and copied into the message by means of the *data request*. The *data request* will include the information that defines whether or not the vehicle is the current initiator of a flow, i.e., *flow initiator*. As mentioned in Section 6.3.1.1, such decision is made upon the expiration of the τ timer. The rationale behind gathering the TrafficMap data only when the vehicle is about to send a TrafficMap Message down to the MAC layer is that during the time a message has to wait before the time scheduled expires, other message could have been received and triggered an alteration of the TrafficMap data in the upper layer. Retrieving the data at this point guarantees that the most up-to-date information available is spread to other vehicles.

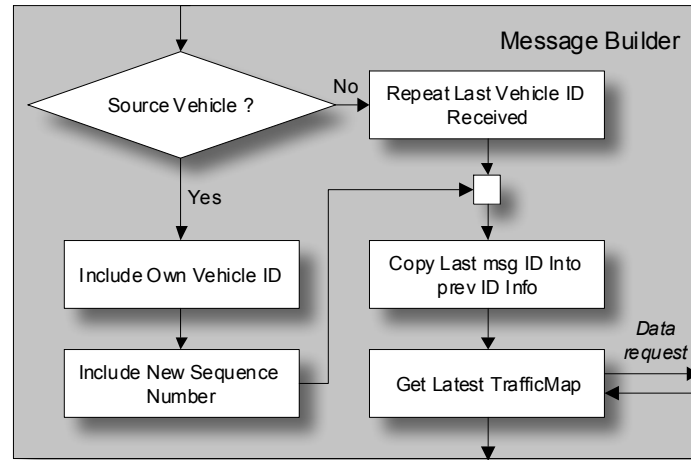


Figure 6.10: The Message Builder

6.4 Time Slot Optimization

The employment of time slots, as described in Section 6.3.1.1, plays a crucial role on the overall performance of the OTHA protocol. The number of time slots defines the maximum time a vehicle will wait before transmitting a message. If such number is high for a sparse network, vehicles might not be equally distributed among the reserved time slots. For instance, we could have a situation with the vast majority of vehicles being assigned to the latest time slot, which would affect negatively the multi-hop delay among vehicles on the road. Finding the most appropriate value for the total number of time slots is not a simple task, though. It requires a global knowledge of the current vehicle density, what is generally not available. One way of providing such density information would be by having all vehicles transmitting a periodical message, i.e., a beacon message. This would probably affect the scalability and performance of the network greatly. An alternative is to rely on traffic flow theory in order to estimate the current density based only on local information as proposed in [Artimy 2007]. However, not every vehicle on the road might be equipped with a radio transmitter and, therefore, the density value estimated would not reflect the actual situation on the road.

Clearly, determining the total number of time slots requires a more detailed study, as a means to accurately estimate the current vehicle density is still an open issue. Nevertheless, a similar problem is also present but in a smaller scale within a pre-defined fixed total number of time slots. Such problem regards the total number of time slots reserved for source and relay vehicles. The number of source vehicles is purely a result of the frequency of speed deviations on the road. Hence, it depends directly on the traffic situation on different areas on the road, e.g., whether there is a free-flow traffic and vehicles are driving at similar speeds or there is a congestion area and the speed profile of the road changes as vehicles arrives in the traffic jam. In this work we propose a time slot optimization that defines dynamically the number of time slots reserved for source and relay vehicles for each vehicle on the road. The underlying assumption is that the current traffic behavior, i.e., whether there are speed deviations on the road, does not

change as frequently as the periodicity of the application defined by the Congestion Assistant by means of the τ timer. For instance, if the periodicity is set to 3 seconds, we argue that within 3 seconds the speed profile of the road will not change considerably in the sense the number of source and relay vehicles will remain approximately the same during this period.

Algorithm 6.1 Dynamic Time Slot Allocation

```

1: if senderVehicleID has not been seen before then
2:   if msg.vehicleID  $\neq$  msg.vehicleID_prev then
3:     neighborList  $\leftarrow$  new entry(vehicleID, sourceVehicle == true)
4:   else
5:     neighborList  $\leftarrow$  new entry(vehicleID, sourceVehicle == false)
6: if timeSlotUpdate timer has expired then
7:    $NS_{source} \leftarrow (neighborList.nrSourceVehicles() / neighborList.size()) * totalNumberOfTimeSlots$ 
8:   if  $NS_{source} == totalNumberOfTimeSlots$  then
9:      $NS_{source} \leftarrow NS_{source} - 1$ 
10:  if  $NS_{source} == 0$  then
11:     $NS_{source} \leftarrow NS_{source} + 1$ 
12:   $NS_{relay} \leftarrow totalNumberOfTimeSlots - NS_{source}$ 
13:  neighborList.clear()
14:  timeSlotUpdate.reset()

```

The method proposed to set the number of time slots for source and relay vehicles dynamically is defined in Algorithm 6.1. The algorithm is executed by each vehicle every time a message is received. In order to identify the sender of a message uniquely, the new field *senderVehicleID* is included in the message structure defined in Section 6.3.2. This is necessary as relay vehicles simply repeat the *vehicleID* information from the last message received. First, if the message received has been sent by a vehicle with an ID that has not yet been seen before, this *vehicleID* is included in the *neighborList* (lines 1-5). The comparison *msg.vehicleID* \neq *msg.vehicleID_prev* verifies whether the message has been originated by a source or relay vehicles. Recall that relay vehicles repeat both *vehicleID* and *vehicleID_prev* fields from the last message received, while source vehicles will have their *vehicleID* values updated to their own identification. This process continues until a certain time limit defined by the *timeSlotUpdate* timer expires (line 6). Since we are interested in capturing the number of source and relay vehicles every time a flow is initiated ahead on the road, we set this timer to the same time defined for the τ timer. The new values for the number of time slots reserved for source (NS_{source}) and relay (NS_{relay}) vehicles are then estimated. In this work, we estimate such numbers by calculating the percentage of source and relay vehicles included in the *neighborList* and then multiplying by the fixed total number of time slots (line 7). A further study is necessary to judge whether relay or source vehicles should be prioritized in this calculation, though. In order to prevent that source or relay vehicles obtain all time slots available, we provide an adjustment to reserve at least one slot for each kind (lines 8-12). Finally, the *neighborList* is cleared and the *timeSlotUpdate* timer is reset to restart the whole process. The result is that each vehicle will have its own estimate in a way the number of time slots utilized is tailored for each area on the road, which might improve the time slot

utilization and consequently the propagation efficiency of the information meant to be delivered along the road. Ideally, vehicles in areas of frequent speed deviations will have an estimate that prioritizes a higher number of time slots reserved for source vehicles.

6.5 Summary of Parameters

<i>Function</i>	<i>Parameter</i>	<i>Description</i>
Time Manager	MIT	Maximum Inter-TrafficMap period, upper bound for the time a vehicle must wait before transmitting
Time Manager	FFP	Flood Free Period, lower bound for the time a vehicle must wait before transmitting
Time Manager	st	duration of one time slot
Time Manager	D_{max}	maximum allowed additional delay within a time slot
Time Manager	NS_{source}	number of time slots reserved to source vehicles
Time Manager	NS_{relay}	number of time slots reserved to relay vehicles

Table 6.1: Parameters defined in the Dissemination Protocol Layer

Performance Evaluation of the OTHA Protocol

Contents

7.1 Evaluation Metrics	87
7.1.1 General metrics	88
7.1.2 Mobility-specific metrics	92
7.1.3 Compression-specific metrics	92
7.2 Simulation Configuration	93
7.3 Static Scenarios	96
7.3.1 Scenario description	96
7.3.2 Results	98
7.4 Mobility Scenarios	108
7.4.1 Scenario description	108
7.4.2 Results	113
7.5 Optimization Methods	118
7.5.1 Compression	118
7.5.2 Time Slot	120

This chapter presents the performance evaluation of the OTHA protocol. The evaluation is performed by means of simulation under static and mobility scenarios.

The chapter is organized as follows: Section 7.1 described the metrics used to evaluate the protocol; Section 7.2 details the parameters values and other aspects of the simulation configuration; Section 7.3 presents the evaluation of the protocol under static scenarios; Section 7.4 presents the evaluation of the protocol under mobility scenarios; and finally, Section 7.5 gives details of the evaluation of the optimization methods proposed for each layer of the OTHA protocol.

7.1 Evaluation Metrics

We detail and motivate in this section the evaluation metrics utilized to measure the performance of the OTHA protocol. For each metric described, we define how its value is calculated for a single simulation run. The mean of all runs for each metric together with its 95% confidence interval are shown in the result graphs. The confidence interval is calculated with the Student's *t*-distribution [Hurst 1995].

7.1.1 General metrics

Reachability

Reachability is of great importance when our goal is to deliver accurate information about the upcoming traffic to every possible vehicle on the road segment that regards the direction of the TrafficMap Message. One possible manner of measuring reachability in a vehicular network is by verifying whether the message has been fully propagated along the road. For instance, on a straight single-lane road that would mean having the message being carried by vehicles from one end to the other, up to the last vehicle present on the road upstream. However, the fact that the end-to-end communication succeeded does not imply that *every* vehicle actually correctly received the message. Because this behavior has been verified during preliminary simulations, we opt by measuring the reachability as the percentage of vehicles in fact correctly received at least one TrafficMap Message per each TrafficMap flow initiated. In addition, in order for such a message to be counted, it has to be considered as relevant by the upper Traffic Filter Protocol Layer. For instance, message originated from vehicles in the opposite direction will not be considered as relevant for the sake of simplification.

The reachability R is then expressed by:

$$R = 100 \times \left(\frac{V_r}{V_t} \right) \times \left(\frac{1}{N_{flows}} \right) \quad (7.1)$$

where V_r is the number of times vehicles received at least one relevant message for all TrafficMap flows initiated, V_t is the total number of vehicles, and N_{flows} is the total number of TrafficMap flows initiated.

Delay

Although the Congestion Assistant is not a time critical application, the information received must be always up-to-date in order to be able to represent the traffic condition ahead accurately. In order to gain some insight in the time the TrafficMap information takes from one extreme of the road to the other, we measure the delay as the time required for the last vehicle present on the road upstream to receive the whole TrafficMap information available. Because the flow started by the *flow initiator* vehicle at one end of the road at time instance zero might be split into multiple TrafficMap information flows along the road (explained in Section 6.2), we consider the arrival time of the last TrafficMap Message received by the last vehicle on the road upstream.

$$D = \frac{T_d}{N_{flowsReceived}} \quad (7.2)$$

where T_d is the total amount of delay considering all TrafficMap flows received in the simulation run and $N_{flowsReceived}$ is the total number of flows received by the vehicle performing the measurement.

System load

The system load indicates how demanded a vehicle is under the utilization of the OTHA protocol, e.g., how often a vehicle is required to participate either transmitting or receiving information required by the protocol. It plays an important role in defining whether other, more critical, applications are still able to propagate their information along with the presence of the OTHA protocol. In order to characterize the overall system load, we measure individually the following metrics: *number of receptions and transmissions*, *overhead*, and *channel utilization*.

Number of receptions and transmissions

The average number of receptions and transmissions per vehicle is simply calculated by counting the total number of messages sent and received during a simulation run and averaged by the number of TrafficMap flows initiated. Because these values are updated upon the transmission and reception of each message in the Dissemination Protocol Layer, it only considers messages correctly received by the MAC layer. The number of transmissions, on the other hand, also take into consideration messages transmitted that may have been lost or collided.

The average number of receptions and transmissions $AN_{receptions}$ and $AN_{transmissions}$, respectively, per vehicle are expressed by:

$$AN_{receptions} = \left(\frac{M_r}{V_t} \right) \times \left(\frac{1}{N_{flows}} \right) \quad (7.3)$$

$$AN_{transmissions} = \left(\frac{M_t}{V_t} \right) \times \left(\frac{1}{N_{flows}} \right) \quad (7.4)$$

where M_r and M_t are the total number of messages received and transmitted, respectively, V_t is the total number of vehicles, and N_{flows} is the total number of TrafficMap flows initiated.

Overhead

The overhead correlates the average number of receptions and transmissions per vehicle described previously. We follow the definition given in [van Eenennaam 2008] and define the overhead O as:

$$O = \frac{AN_{transmissions}}{AN_{receptions}} \quad (7.5)$$

Normally, there will be a higher number of receptions when compared with the number of transmission, since one transmission might yield the reception in various vehicles. Thus, when the average number of transmissions is greatly higher than the average number of receptions, we say the overhead is low.

Observed channel utilization per vehicle

The channel utilization measures the percentage of total simulation run time spent by each vehicle on average receiving and transmitting TrafficMap Messages during the propagation of a certain information flow started by a *flow initiator* vehicle. As a result, this value also provides an important estimate of the amount of time left for other more critical applications to utilize the channel. Differently from the average number of receptions and transmission described previously, the channel utilization takes into account any noise detected by a vehicle, either correctly received messages or simply errors or collisions during message receptions. Both values are measured on the physical layer of the IEEE 802.11p implementation utilized in our simulation.

Thus, the total channel utilization per vehicle CU is defined by:

$$CU = 100 \times \left(\frac{T_t + T_r}{V_t} \right) \times \left(\frac{1}{T_s} \right) \quad (7.6)$$

where T_t , T_r , and T_s are the average time spent transmitting and receiving TrafficMap Messages, and the total simulation run time, respectively.

Slot utilization

The slot utilization is an important metric that indicates how well the time slots defined by our suppression technique (described in Section 6.3.1.1) are assigned to vehicles. The performance of the Dissemination Protocol Layer depends directly on the slot distribution among vehicles in each broadcast. For instance, a higher utilization of time slots concerning relay vehicles positioned further in the message direction implies that the TrafficMap information has been propagated efficiently. We calculate the Slot Utilization (SU_i) as the percentage number of times each time slot i is utilized by vehicles in transmissions that in fact occur (scheduled + transmitted), during a simulation run.

$$SU_i = 100 \times \left(\frac{TS_i}{TS_{total}} \right) \quad (7.7)$$

where TS_i and TS_{total} are number of times a time slot i is utilized and the total number of times all time slots together are utilized.

Accuracy

The prime goal of the OTHA protocol is to provide an accurate view of the traffic ahead to vehicles. The accuracy of the information contained in the entries added to the TrafficMap is directly influenced by the following factors:

- Thresholds defined in the sensitivity ε function in the upper Traffic Filter Layer, which decides whether the speed deviation of two entries are high enough to include a new entry to the TrafficMap.
- Errors occurred during the propagation of TrafficMap Messages in the vehicular network. Since TrafficMap Messages may be lost or collided during its transmission, some informa-

tion may be lost and not included to the TrafficMap. The result is that not every speed deviation detected is reported in the TrafficMap, increasing inaccuracy.

- The mobility of vehicles together with the propagation delay of TrafficMap Messages may influence on how up-to-date the information received is. For instance, if the complete information needed a couple of seconds to travel from a flow initiator vehicle up to another vehicles 10 km upstream, the information may not be completely up-to-date any longer, as the speed and position of vehicles are no longer the same.

The way we evaluate accuracy is by measuring the error (difference in speed values) of the data collected, i.e., the speed values of the entries added to the TrafficMap, compared with the real speed of vehicles present on the road, as proposed and employed in [van Eenennaam 2008]. The error measured will include inaccuracies caused by every factor outlined above.

The method utilized begins by interpolating a line between every pair of (position, speed) points contained in the entries added to the TrafficMap. In the sequel, the error is calculated by measuring the distance between speed values of the real points (actual position and speed of vehicles on the road) and their projected speeds for the same position in the interpolated line.

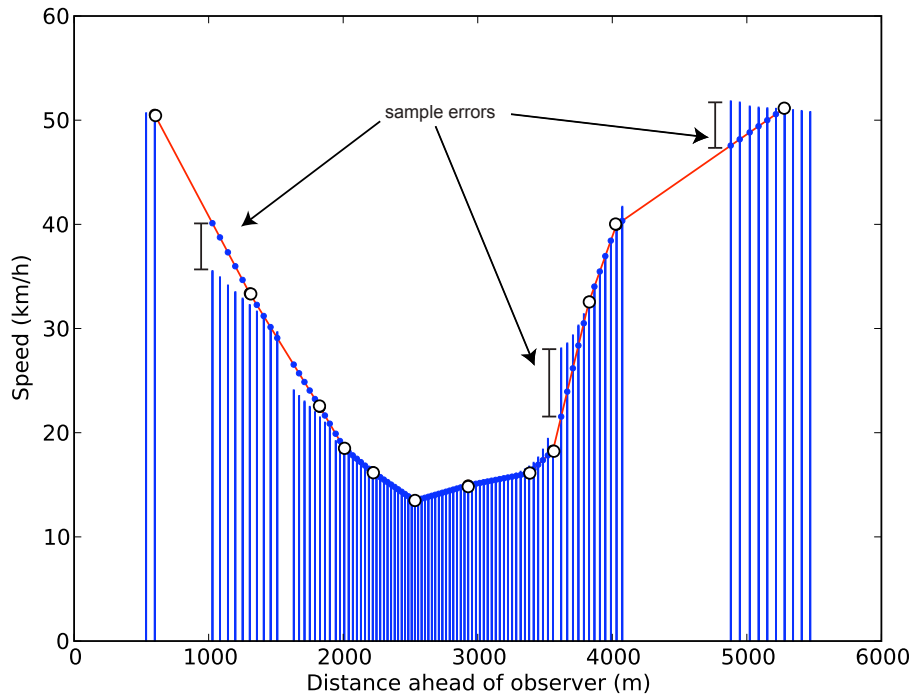


Figure 7.1: The sampling error calculation method

Figure 7.1 illustrates how the measurement is done. In this example, the points (position, speed) of the TrafficMap entries are depicted by the empty (white) circles. For every pair of points formed by subsequent points from lower to higher positions in the figure, a line is interpolated. The dark points over each line represent the projection of the speed values for

the real position of vehicles. The difference between the projected and the real speed values are referred to as *sample errors*. The total sampling error (SE) is then calculated as follows:

$$SE = \left(\frac{E_t}{P_t} \right) \quad (7.8)$$

where E_t is the sum of all existing sample errors and P_t is the total number of points projected for all TrafficMaps received in a simulation run.

7.1.2 Mobility-specific metrics

Distance of awareness

Since the static scenarios are meant to evaluate the OTHA protocol in a controlled environment, the connectivity among vehicles, i.e., whether they are able to reach each other by means of a multi-hop communication, is known beforehand. The mobility scenario considered in this work, on the other hand, is meant to evaluate the system in a more realistic environment and includes moments where a complete connectivity of vehicles is not present. In such scenario, in order to evaluate how well the OTHA protocol performs with regard to the propagation of TrafficMap information along the road, we include a new metric that evaluates the distance of awareness achieved in each time a TrafficMap is received in a simulation run. For each TrafficMap received, the distance of awareness (DA) obtained is calculated as follows:

$$DA = Distance(P_{max}, P_{ov}) \quad (7.9)$$

where the function *Distance* calculates the distance between the TrafficMap entry points P_{max} and P_{ov} that contain the (x,y) position values. P_{max} regards the oldest entry added to the TrafficMap. Because the traffic information is propagated upstream, such entry will regard the furthest position with regard to the position P_{ov} of the observer vehicle calculating the awareness. In our simulations, we choose a static vehicle to behave as the mentioned observer vehicle to receive and calculate the total awareness achieved.

In order to understand the real performance achieved by the OTHA protocol we compare the observed distance of awareness with the maximum theoretical distance of awareness estimated for each time instance of a simulation run. Such estimate is defined as the maximum possible distance that can be achieved from the point of view of the observer vehicle by means of multi-hop communication, considering a theoretical transmission range of 250 meters for each vehicle. This is illustrated in Section 7.4.

7.1.3 Compression-specific metrics

In order to evaluate the performance of the compression optimization method proposed in Section 5.4, we include additional metrics, namely, the *percentage of compression* and the *TrafficMap size*.

Percentage of compression

The percentage of compression (PC) regards the TrafficMap data in bytes compressed compared with the raw uncompressed data and is defined as follows:

$$PC = 100 - \left[\left(\frac{D_{compressed}}{D_{raw}} \right) \times 100 \right] \quad (7.10)$$

where $D_{compressed}$ is the total amount of bytes compressed and D_{raw} is the total amount of bytes of uncompressed TrafficMap data, both with regard to a simulation run.

TrafficMap size

The TrafficMap size is meant to complement the previous metric by giving an insight of the actual number of bytes needed for the representation of the road in the TrafficMap with and without compression. Its calculation is straightforward as we simply calculate the total average number of bytes used for the construction of each TrafficMap in a simulation run.

7.2 Simulation Configuration

The simulation is carried out by means of the OMNET++ simulator version 4.0 [Varga *et al.* 2001]. Our choice is based upon the great flexibility offered by the simulator with regard to the possibility of designing multiple network models and topologies. The current version 4.0 also provides an integration with the Eclipse IDE [Foundation 2009], which facilitates the development and testing of the protocol designed. Further details on the comparison of OMNET++ with other simulators such as NS-2 [McCanne *et al.* 2000] can be found in [Xian *et al.* 2008, Orfanus *et al.* 2008]. In our simulation, we utilize the Mobility Framework available in [Drytkiewicz *et al.* 2003] and adjust the available implementation of the IEEE 802.11b protocol to meet certain basic characteristics of the 802.11p version, namely, frequency band, bandwidth, bit rate, and other parameters that we describe in this section.

The parameters utilized for the simulations considered in this work are detailed in Table 7.1. In the Traffic Filter layer of the OTHA protocol, further study on traffic theory is required to determine the optimum values for the parameters utilized. The choice we make is based on a conservative approach and on previous successful results obtained in [van Eenennaam 2008]. In the averaging function, the Δ parameter is set to 500 meters and α to 1. We assume that after 500 meters the speed of vehicles are not representative any longer to averaged to the last entry added to the TrafficMap. We also consider that the weight given to the averages performed by vehicles decreases linearly with the distance, thus, $\alpha = 1$. For the sensitivity ε function, the d threshold set to 1000 meters implies that a new entry is added by vehicles whenever their distance to the position value of the last entry added to their lanes in the TrafficMap exceeds 1 km. This guarantees a refresh of information along the road in every kilometer. The o_{slope} and p_{slope} are set to 0.933 and 0.888, respectively. The reasoning behind a greater value assigned for o_{slope} is that from a safety point of view, the braking of vehicles ahead on the road is a more crucial knowledge to vehicles behind than vehicles accelerating. Moreover, the

Congestion Assistant requires more detailed information regarding vehicles arriving a traffic jam, as described in Chapter 5. The same reason applies for the o_{offset} and p_{offset} values, which are set to 5 and 7, respectively. These values prevent that small speed variations present in traffic jams induce the inclusion of a new entry in the TrafficMap. In the Reduce TrafficMap function, we assign the area of $10 \times 10 \text{ km}^2$ for the road segment. This area is meant to cover every geographical part of the simulation scenarios. The following values for this layer regard the entry size and the maximum size for the TrafficMap structure. The former is set to 14 bytes that comprises the following information: 2 bytes for speed (to include speeds greater than 255 km/h) + 8 bytes for the geographic position (as suggested in [Davis *et al.* 1996]) + 4 bytes for the timeStamp (according to the UNIX standard [Stevens & Rago 2005]). The latter value is set to 2312 bytes which is the maximum payload size allowed for a MAC frame defined in the IEEE 802.11 protocol [Gast & Loukides 2002]. The last value for this layer is the d threshold used by the Greedy approach, the method chosen in this work to compress the TrafficMap data (see Section 5.4). We set it to 250 meters, what implies that within the range of 250 meters the position values of two different entries may be considered as similar. This based is backed by the fact the OTHA protocol regards the awareness of long distances up to tens of kilometers and the difference of 250 meters between two entries may still represent the speed pattern of the road correctly.

In the Dissemination Layer, we set the parameters based on preliminary simulations performed during the development of this work and on previous results obtain in [van Eenennaam 2008, Wisitpongphan *et al.* 2007]. In particular, the number of time slots influences greatly on the overall performance of the system, since it defines the time waited by vehicles before sending a TrafficMap Message. Ideally, such value could be determined dynamically as mentioned in 6.4. In this work, we limit ourselves to static values and set the NS_{source} and NS_{relay} to 2 and 5, respectively. The reason why we assign two time slots for source vehicles is to avoid that many vehicles assigned to a single time slot attempt to transmit at the same time and cause collisions. During a preliminary phase of our simulation experiments, it has been noticed that due to the adding of an entry every time the d threshold in the sensitivity ϵ function is exceeded, many source vehicles nearby tried to broadcast almost simultaneously, which compromised the reachability of the system. The choice of five time slots reserved to relay vehicles is supported by the fact that good results have been achieved in simulations performed in [Wisitpongphan *et al.* 2007]. Moreover, it is reasoned that a greater number of relay vehicles would normally be present on the road when compared with source vehicles, given that the presence of source vehicles are only due to speed deviations captured by the thresholds defined in the sensitivity ϵ function in the upper layer. Differently, relay vehicles are constantly needed to simply propagate the TrafficMap information upstream. Still with regard to our suppression technique, the st (slot time) and the D_{max} values are set to 0.009 and 0.0029 seconds, respectively. It has been found in [Wisitpongphan *et al.* 2007] that 0.005 gives enough time for vehicles assigned to later time slots to receive an echo of the message scheduled and cancel their transmissions. In order to guarantee that the value assigned to D_{max} (0.0029) does not overlap other time slots and still gives time to vehicles in later slot to cancel their messages, we overestimate the calculation $0.0029 + 0.005 = 0.008$ and set st to 0.009 seconds. The D_{max} in turn is calculated as a function of the available DIFS period utilized in 802.11 protocols. We consider the following relation $D_{max} = 50 \times DIFS$.

<i>Protocol Layer</i>	<i>Function</i>	<i>Parameter</i>	<i>Value</i>
Traffic Filter	Averaging function	Δ	500 m
Traffic Filter	Averaging function	α	1
Traffic Filter	Sensitivity ε function	d threshold	1000 m
Traffic Filter	Sensitivity ε function	o_{slope} threshold	0.933
Traffic Filter	Sensitivity ε function	p_{slope} threshold	0.888
Traffic Filter	Sensitivity ε function	o_{offset} threshold	5
Traffic Filter	Sensitivity ε function	p_{offset} threshold	7
Traffic Filter	Reduce TrafficMap function	road segment	10 x 10 km ²
Traffic Filter	TrafficMap structure	entry size	14 bytes
Traffic Filter	TrafficMap structure	max structure size	2312 bytes
Traffic Filter	Similar function (compression)	d threshold	250 meters
Dissemination	Time Manager	MIT	3.0 s
Dissemination	Time Manager	FFP	0.1 s
Dissemination	Time Manager	st (slot time)	0.009 s
Dissemination	Time Manager	D_{max}	0.0029 s
Dissemination	Time Manager	NS_{source}	2
Dissemination	Time Manager	NS_{relay}	5
MAC 802.11p	Backoff algorithm	slot time	13 μ s
MAC 802.11p	Backoff algorithm	SIFS	32 μ s
MAC 802.11p	Backoff algorithm	DIFS	58 μ s
MAC 802.11p	Backoff algorithm	min CW value	15
MAC 802.11p	Backoff algorithm	max CW value	1023
MAC 802.11p	Basic configuration	bit rate	6 Mbit/s
PHY 802.11p	Basic configuration	frequency band	5.87 GHz
PHY 802.11p	Basic configuration	bandwidth	10 MHz
PHY 802.11p	Basic configuration	tx power	168.98 mW

Table 7.1: Parameters utilized in the simulations

The MIT is set to 3 seconds for the reasons explained in Section 6.3.1.1. Regarding the FFP period, further study must be done in order to evaluate and find a proper value for it. We motivate in Section 6.3.1.1 that such value should fall in between the time utilized by the time slots in the suppression technique and the MIT value. High values may imply long delays to receive the complete information started by the flow initiator, since a flow may be split during its propagation along the road, as explained in Section 6.2. On the other hand, high values may contribute to merge multiple flows into a single one, what could diminish the number of messages and consequently the load in the network. In our simulations we set FFP to 0.1 seconds, which is a value considerable lower than the MIT set and at the same time it may give the opportunity

for multiple flows to merge, since it is a value greater than the maximum time waited by a vehicle during our suppression technique $((0.009 \times 7) + 0.0029 = 0.0659)$.

The remaining parameters regard the IEEE 802.11p protocol utilized as the MAC and PHY layers below the OTHA protocol. The parameters defined in our simulations are based on latest drafts of the protocol described in [IEEE 2006, Jiang & Delgrossi 2008] for the class with the lowest priority in the Enhanced Distributed Channel Access (EDCA) Quality of Service (QoS) extension provided by IEEE 802.11e [Suthaputehakun & Ganz 2007]. The parameters used by the backoff algorithm, namely, the slot time, SIFS, DIFS, min and max contention window (CW) sizes, are set to 13 μ s, 32 μ s, 58 μ s, 15 and 1023, respectively. Although the bit rate is defined as varying from 6 to 27 Mbit/s, we choose the lowest value 6 Mbit/s as a worst-case scenario regarding transmission delay. The frequency band utilized is 5.87 GHz, which is one the service channels reserved in the DSRC spectrum band, as described in Section 2.2. The bandwidth is set to 10 MHz as assigned to each channel in the mentioned spectrum. Furthermore, the transmission power is set to 168.98 mW. This value has been previously used in [van Eenennaam 2008] as an estimate to achieve 500 meters of interference range and 250 of transmission range, since it is assumed that the interference range is generally twice the transmission range [Xu *et al.* 2002]. This power level value has been derived from the Friis Free Space propagation model [Friis 1946], which is the model utilized by the Mobility Framework for OMNET++. The employment of this model implies that only vehicles can cause interference in the environment considered, which is a rather simplistic assumption, since in order to consider more realistic scenarios one must consider the existence of multipath and reflection effects incurred, for instance, by the presence of buildings. We argue, however, that the propagation model may suffice for a first evaluation of the OTHA protocol. The choice of 250 meters of transmission range also is made to consider a worst-case scenario, since in latest 802.11p drafts the transmission range is expected to achieve up to 1 km.

7.3 Static Scenarios

7.3.1 Scenario description

In order to evaluate the performance of the OTHA protocol in every scenario addressed in this work, namely single-lane road, multiple-lane road, road with junctions, and road with multiple directions, we define one basic static scenario for each of them, as illustrated in Figure 7.2. In each scenario depicted, the arrow outside the road indicates the direction of vehicles on the road. The dark circle represents the *flow initiator* vehicle and the arrow next to it indicates the message direction of the TrafficMap information flow. All flow initiators are placed on the extreme side of each road towards the direction of vehicles. In the junction and road with opposite directions scenarios there are two flow initiators. In the junction scenario there is one flow initiator for each road. Similarly, in the opposite direction scenario one flow initiator is placed in each direction of the road. Moreover, the vehicle represented with a white circle placed in the extreme opposite end of the road when compared with the flow initiator, is used to gather relevant information for a few metrics, namely, end-to-end delay and accuracy. Specifically, the calculation of accuracy relies on the TrafficMaps received by this vehicle.

For each scenario, we perform 50 simulation runs for each of the following vehicle densities:

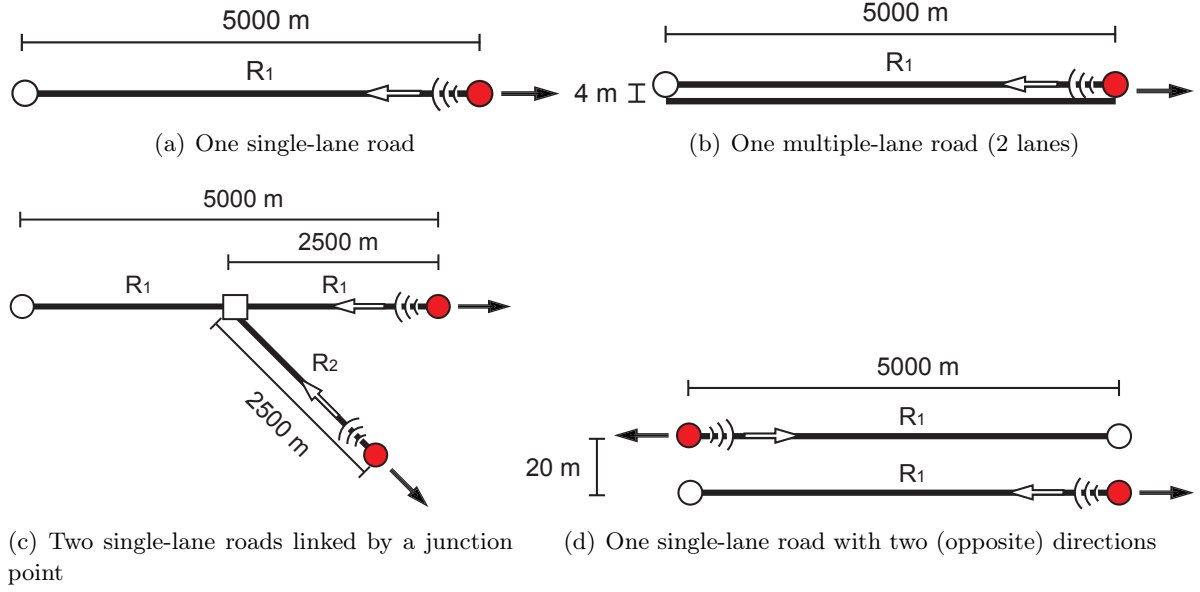


Figure 7.2: Illustration of the static scenarios considered

20, 40, 60, 80, and 100 vehicles/km/lane. The *flow initiator* vehicle creates the TrafficMap structure and starts transmitting at time instance zero. In each simulation run, 50 flows are initiated by this vehicle in intervals of 3 seconds, totalizing in a simulation run of 150 seconds. As a result, for every density of each scenario considered $50 \times 50 = 2500$ flows are initiated. Since vehicles do not move, internal events defined in the upper Traffic Filter Protocol Layer that concern change in speed, lane or road, are never executed during the entire simulation.

In order to evaluate the OTHA protocol in scenarios similar to what is found in real roads, the distribution of vehicles along the road is determined by the Intelligent Driver Model (IDM) described in [Treiber *et al.* 2000]. The IDM is a continuous car-following model which is essentially defined by an acceleration function. The vehicle distribution utilized in every scenario and density is illustrated in Appendix A. All scenarios illustrated are basically snapshots containing the speed and position of vehicles taken after a random time the IDM model is executed. The steps taken to generate each snapshot are the following. The IDM model is applied in a uniform distribution of vehicles and as the time evolves the speed of vehicles is adjusted according to their desired speed and the speed of the vehicles right in front, among other parameters such as maximum acceleration allowed. In every scenario a traffic jam is induced by determining a lower maximum speed value in a certain region of the road. We choose different snapshots of vehicle distributions generated by the IDM model for scenarios where there are independent parts on the road, namely, the two roads of the junction scenario and the two lanes on the multiple-lane scenario. For the opposite direction scenario even though both directions have exactly the same snapshot, the starting point of each flow are opposite to each other, which results in two completely different information flows. The main purpose is to simulate situations that can be present in real roads.

As we evaluate the OTHA protocol in different and independent scenarios, we are not able to draw conclusions directly from the comparison of results (values) between them. The main purpose is to evaluate the behavior of the OTHA protocol in realistic scenarios of each type separately and generate conclusions upon their evolution as the vehicle density increases. In particular, we want to find out whether the protocol functions correctly in all of them for every metric. The comparison between scenarios that we sometimes provide is limited by what it is generally expected and what has been achieved.

7.3.2 Results

In this section we present the results for the static scenarios considered in this work. In order to provide examples of how the TrafficMap structure has been generated in our simulations we place one TrafficMap output for each static scenario in Appendix B.1.

7.3.2.1 Reachability

The performance of the OTHA protocol for each scenario with respect to reachability is depicted in Figure 7.3. At density 20 vehicles/km/lane, generally the reachability is poor for every scenario due to the sparse vehicle network, and therefore, lack of connectivity among vehicles present on the road. In the multiple-lane scenario, oppositely, the reachability achieves a high mark of almost 100%. This is due to the fact that the vehicle distribution for each lane is different and thus the lack of connectivity in one lane is compensated by the other.

As of 40 vehicles/km/lane, in the single-lane, junction, and opposite direction scenarios the reachability remains to a constant of almost 100%. In particular, the results for the opposite direction scenario has been better than our pessimistic exceptions. Due to lack of coordination between the flows started at the same time on opposite ends of the road, we envisioned that the encounter of these flows would yield a high level of transmission collisions and errors. Apparently, the fact that these flows interfere with each other only during their meeting, which is in the fraction of milliseconds, even with the probability of collision being higher some vehicle in each direction could transmit and continue with both flows.

The reachability in the multiple-lane, on the other hand, is highly affected and decreased down to 80% as the density increases. This can be explained by the fact that the number of vehicles within the transmission range is the double when compared with the single-lane scenario, which results in the double of vehicles on average within each time slot assigned to vehicles. Hence, the number of collisions and errors are expected to increase considerably due to a high probability of multiple vehicles attempting to transmit simultaneously.

7.3.2.2 Delay

Figure 7.4 depicts the performance results for the multi-hop delay from one end to the other of the road. At density 20, only the multiple-lane scenario is illustrated, since it is the only scenario with a complete end-to-end connectivity among vehicles.

The delay in the opposite direction, single-lane, and junction scenarios behaves similarly with a smoothly decrease throughout the increase of density. A higher delay in low density situations is expected, since the time slots utilized by our suppression broadcast method may

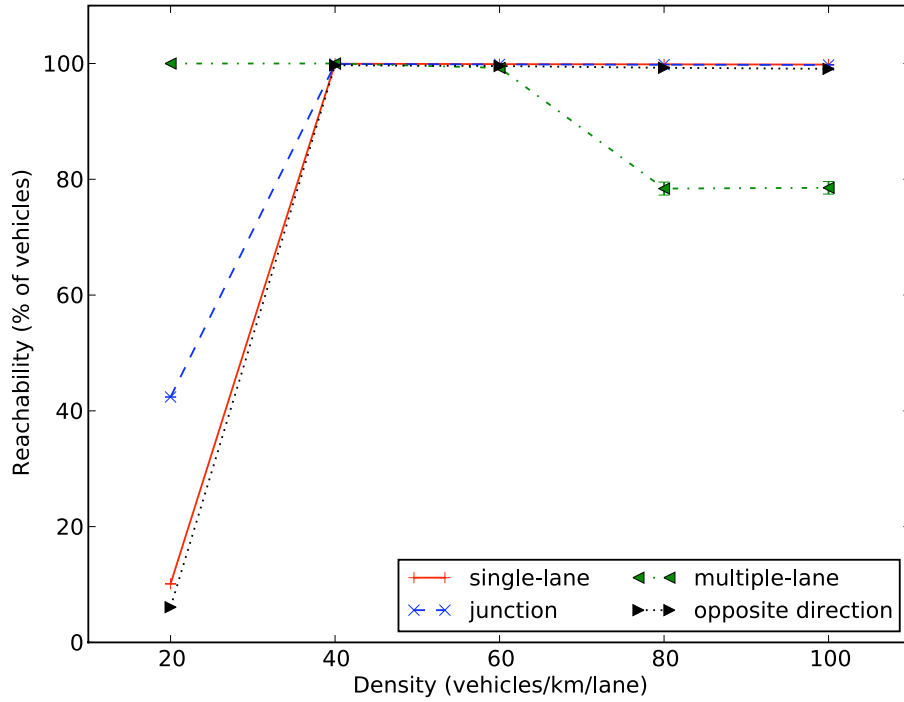


Figure 7.3: Reachability x density

not be equally distributed among vehicles. In fact, in such densities it is possible that there only relay vehicles and yet positioned close to the sender. The result is that long delays will be assigned to transmission of vehicles due to a higher time slot number. As of density 60, on the other hand, vehicles are likely to be well distributed along the road and thus with a higher chance of existing vehicles assigned to early time slots.

As it has occurred in our simulations, it is expected that the delay will generally be higher in junction scenarios when compared to single-lane scenarios, for instance, due to the existing two TrafficMap flows started in different roads. Since it is likely that these flows will arrive in the opposite end of the scenario asynchronously, later flows might be delayed by the FFP timer set to vehicles during the travel of the first flow. As explained in Section 6.3.1.1, this timer is set to guarantee a certain time interval between consecutive transmissions. As a consequence, the delay value which is defined by the arrival time of the last relevant TrafficMap Message received will be higher.

The OTHA protocol has its worst performance in the multiple-lane scenario with regard to the end-to-end delay. The high number of vehicles per time slot and the consequent high probability of transmission collisions and errors might be the main causes of the increase in delay in high densities. For instance, it could happen that many vehicles attempt to transmit in early time slots and have their transmissions collided. Vehicles with transmissions assigned to later time slots may have the chance to transmit then. In addition, similarly to what has been explained to junction scenarios, because of the splitting of flows started by the *flow initiator* vehicle, the FFP timer also induces a higher delay in the end-to-end propagation of TrafficMap

information flows.

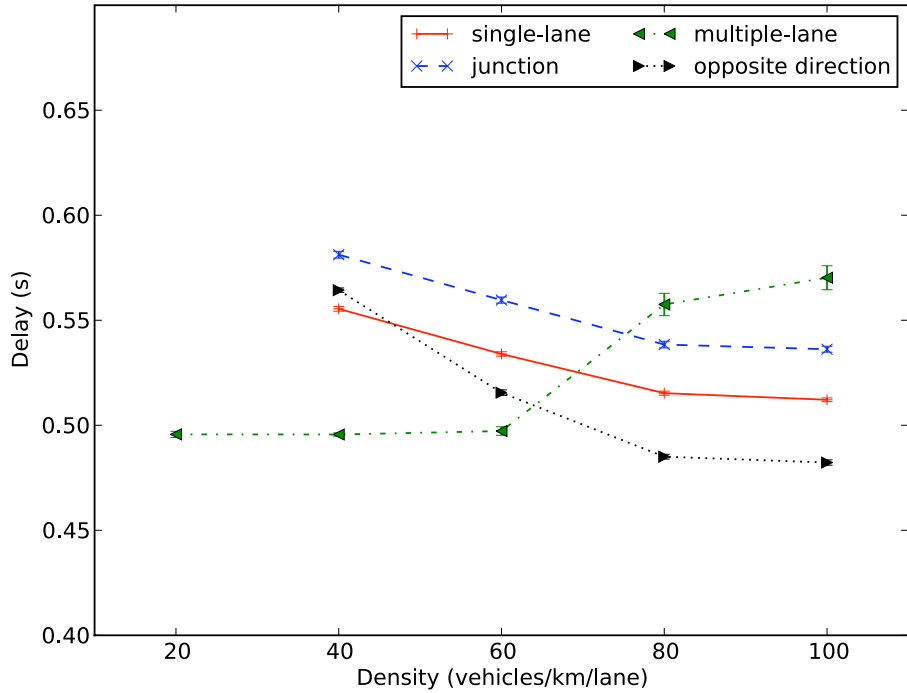


Figure 7.4: Delay x density

Despite the differences in the delay evolution along the increase of density between the scenarios considered, the delay has been always around between 0.5 and 0.6 seconds, what preserves the freshness of the traffic information disseminated in the order of milliseconds.

7.3.2.3 System load

As explained in Section 7.1, the overall system load is evaluated by means of four metrics, namely *number of receptions*, *number of transmissions*, *overhead*, and *channel utilization*.

The average number of receptions per vehicle is illustrated in Figure 7.5. The single-lane, junction and opposite direction scenarios have a similar behavior regarding this metric as the density is increased. In such scenarios the number of receptions increases with density, a fact that is explained by the higher number transmissions regarding a single time slot. Since transmissions scheduled for a common time slot cannot cancel each other, more vehicles nearby results in more transmissions. This can be verified in Figure 7.6 where the average number of transmissions per vehicle is depicted. After a small decrease between densities 40 and 60, the number of transmissions is smoothly increased throughout higher densities.

As it has occurred in our simulations, we expect that the number of receptions per vehicle in the single-lane scenario will be the lowest among the scenarios considered because of the least number of vehicles and TrafficMap information flows existing on the road. In junction and opposite direction roads, due to the existence of more than one flow, the number of reception will be probably higher when compared with single-lane roads. In particular, the opposite direction

road scenario is expected to present the highest value among the three scenarios mentioned. Since flows starting from opposite directions will go through the whole road, most vehicles will receive TrafficMap Messages from their own flow and from the flow coming from the opposite direction.

The multiple-lane scenario presents the worst performance in terms of number of receptions and transmissions. Once again, because of its high vehicle density within a single time slot, both number of receptions and transmissions are high under low densities but it quickly decreases with high densities as collisions and errors are more frequent. This is simply supported by the fact that reachability is decreased in these densities and thus fewer vehicles send or receive TrafficMap Messages.

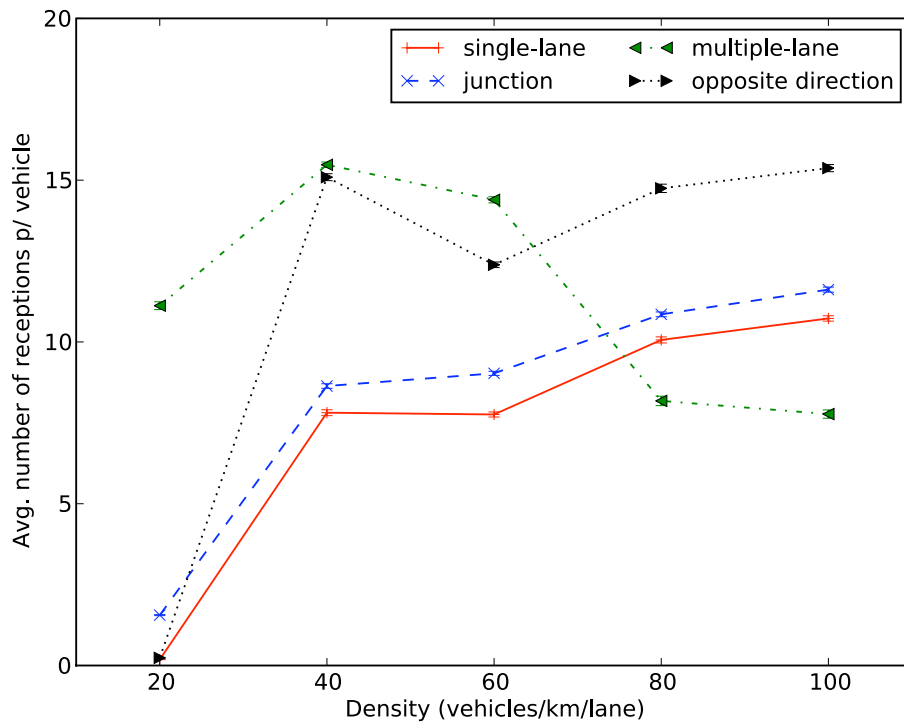


Figure 7.5: Number of receptions x density

In particular, the results presented for the number of transmissions show that on average a vehicle send less than one message in each scenario considered in our simulations.

The overhead correlates the average number of receptions and transmissions per vehicle. As illustrated in Figure 7.7, as of density 40, the junction and single-lane scenarios behave similarly. Since their values for receptions and transmissions maintain a proportional difference, their overhead ratio becomes near each other. The opposite direction scenario presents the lowest overhead rates for every density, which is influenced by the fact that the number of reception is greatly increased by the presence of a flow traveling in the opposite direction. In all these scenarios, the overhead ration is almost constant. Differently, the overhead ratio for the multiple-lane scenario increases with density probably due to reasons previously mentioned such as collisions and transmission errors.

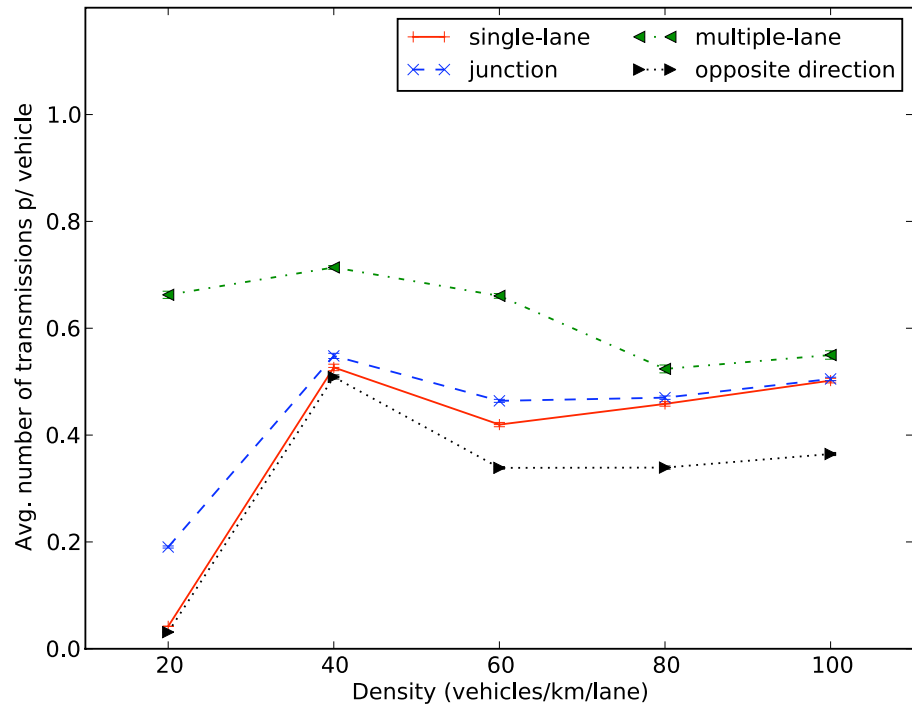


Figure 7.6: Number of transmissions x density

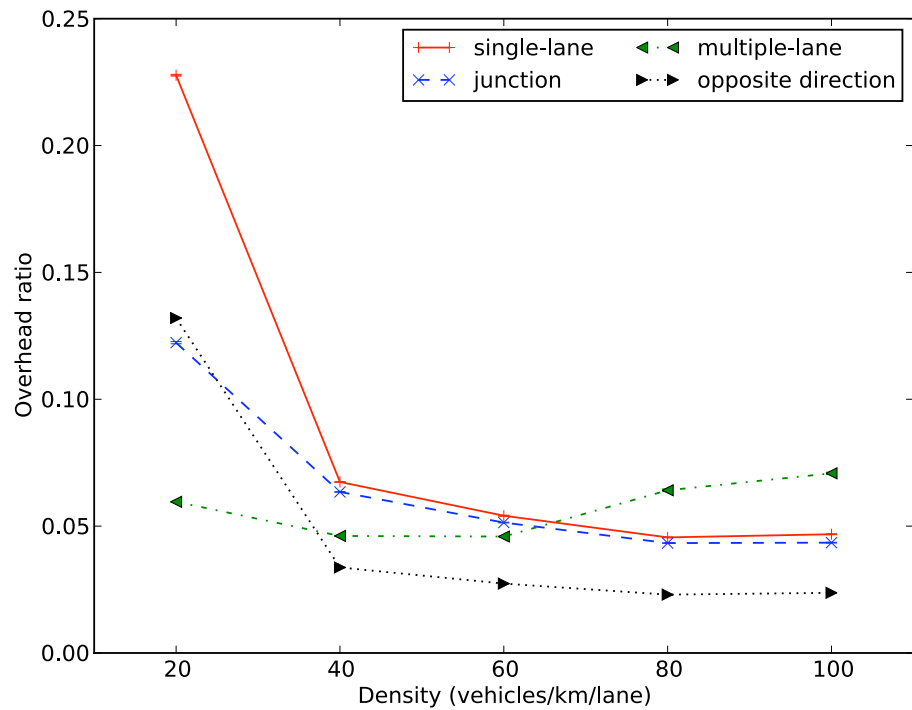


Figure 7.7: Overhead x density

The last metric evaluated to assess the overall system load is the channel utilization per vehicle during the period of time considered. For static scenarios we evaluate the channel utilization for each traffic flow period determined by the MIT parameter, i.e., 3 seconds. Figure 7.8 shows that generally the channel utilization increases with density. This is expected for the same reasons discussed previously for the number of receptions and transmissions, i.e., more vehicles within a single time slot results in more transmissions and thus more receptions.

As it has occurred in our simulations, we expect that the single-lane scenario will present the lowest channel utilization for every density. Because there are fewer vehicles within each time slot on average, vehicles will spend less time transmitting and receiving messages on the road. The remaining scenarios suffer from the existence of multiple flows on the road, either from the opposite direction, other roads (junction), or splitting of a flow into multiple ones (multiple-lane).

An important observation one can make from these results is that, even considering multiple-lane scenarios which have demonstrated to have a poor performance when compared with the remaining scenarios, the channel utilization remained less than 0.7% of the total simulation run time (vehicles were idle 99.3% of the time). Therefore, it is reasonable to argue that other applications will have a high probability of delivering their information under the presence of the OTHA protocol for the scenarios considered in this section.

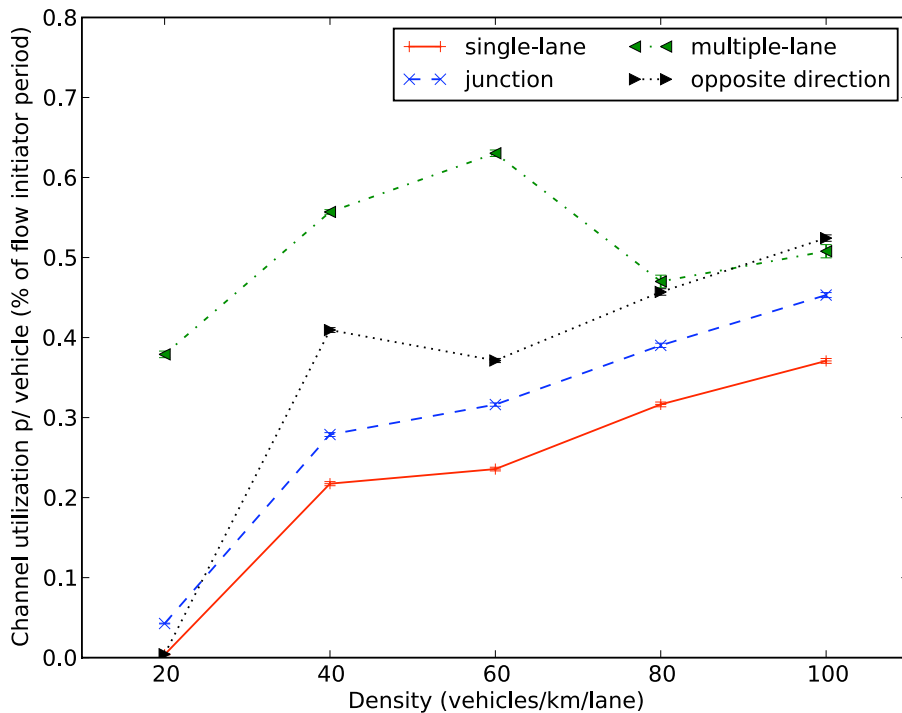


Figure 7.8: Channel utilization x density

7.3.2.4 Slot utilization

The slot utilization distribution reflects directly on the multi-hop delay needed for the propagation of TrafficMap information flows inserted by the OTHA protocol. Ideally, transmissions would only take place on vehicles positioned either in the earliest time slot reserved for relay vehicles, namely slot 2, or on source vehicles in slot 0. However, collisions, errors, and the non-presence of vehicles assigned to a certain time slot due to sparse networks, invalidate this assumption.

The time slot distribution is illustrated in Figure 7.9 with the confidence intervals illustrated on the top of each individual bar destined for each scenario considered. In (a) the utilization percentage of early time slots, namely, 0, 1 and 2 are evidently higher than other time slots, achieving up to 90% in high densities. The non-utilization of slots reserved to source vehicles 0 and 1 for density 20 is simply due to the presence of a sparse network, where no messages have been received by any source vehicle, i.e., the flow stopped before any source vehicle received a TrafficMap Message. Similar results can be observed for the opposite direction scenario. In fact, despite the existence of opposite flows traveling along different directions, the scenario is basically two independent single-lane roads. Differently, the multiple-lane and junction scenarios ((b) and (c)) present a general lower utilization of early slots, namely, 0, 1 and 2. The same explanation given previously for higher delays is argued to be valid for the latter scenarios. The lower utilization of these time slots is influenced by the FFP timer, as multiple flows arrive asynchronously in the last vehicle located at the opposite end of the road with respect to *flow initiator* vehicles. Since the time difference between multiple flows are likely to be in order of a few milliseconds, as taken for a message transmission, vehicles which have priorly used early time slots will now have to wait the FFP timer before they can transmit again. Therefore, vehicles which have not transmitted before because they have been assigned to later time slots will now have the chance to transmit. The result is a higher utilization of later time slots such as slots 3 and 4.

In fact, the relation between the assignment of time slots and the end-to-end delay performance of the OTHA protocol is evident when both results are compared. Similarly to what has been shown in Figure 7.4 for the delay evaluation, multiple-lane and junctions are slightly affected with an increase in delay. As mentioned here, such result is a direct consequence of the higher utilization of later time slots.

7.3.2.5 Accuracy

Accuracy is a crucial measure to be evaluated in our protocol. If the information arrived to a vehicle does not contain a good representation of the road ahead, the radio channel resources have been pointlessly wasted. We evaluate accuracy by measuring the amount of error contained in the TrafficMap entries added along the road.

Figure 7.10 provides the results for the sampling representation error in km/h for each scenario. The sampling error changes similarly to every scenario considered. As the density is increased, the sampling error decreases simply because vehicles at such high densities drive at considerably low speeds due to the occurrence of traffic jams and therefore the speed deviation present on the road is also lower. For the same reasons as mentioned previously, since an end-to-end connectivity has been present only for the multiple-lane scenario in density 20, the sampling

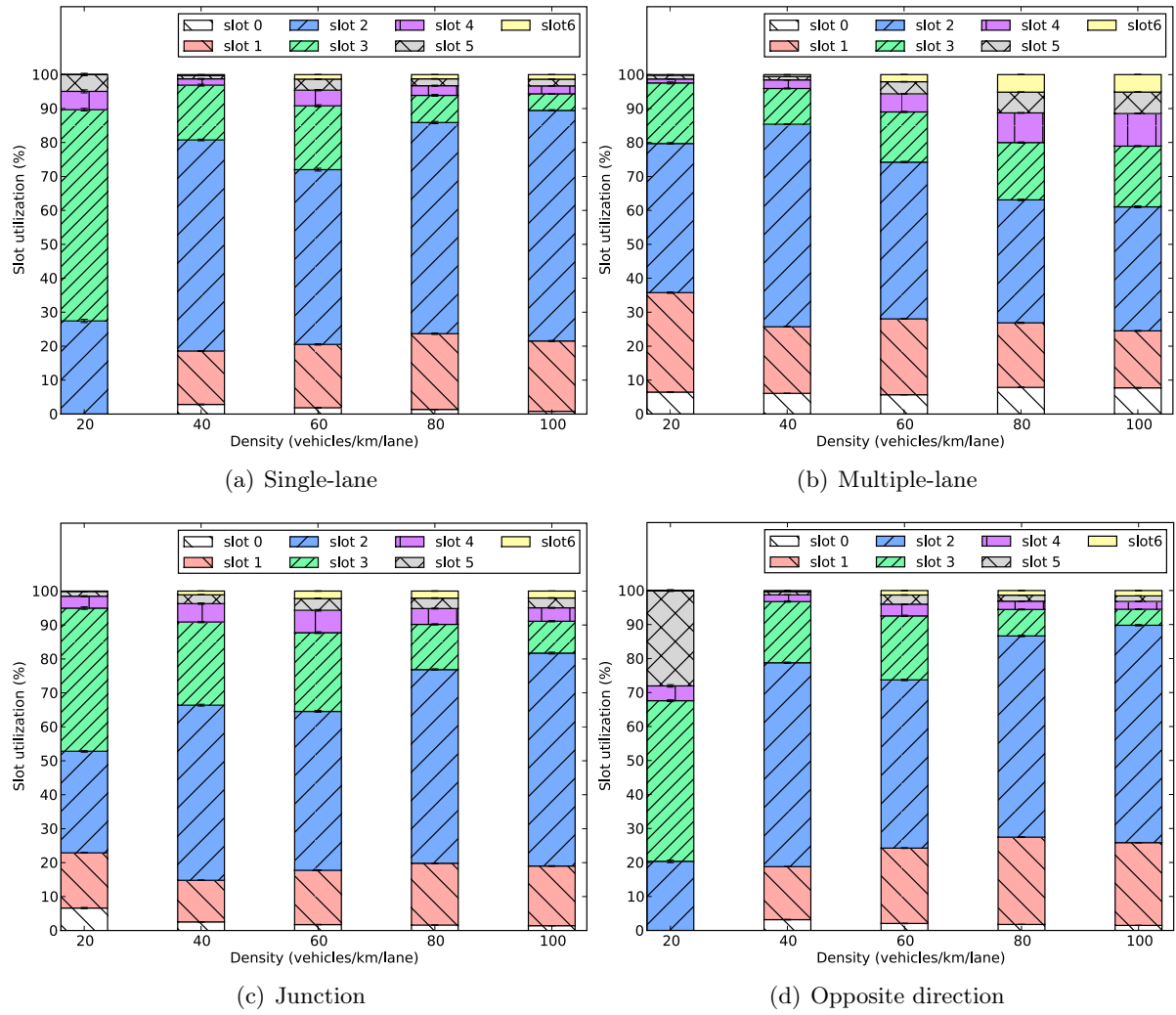


Figure 7.9: Slot utilization x density

error has not been included for the remaining scenarios in this figure.

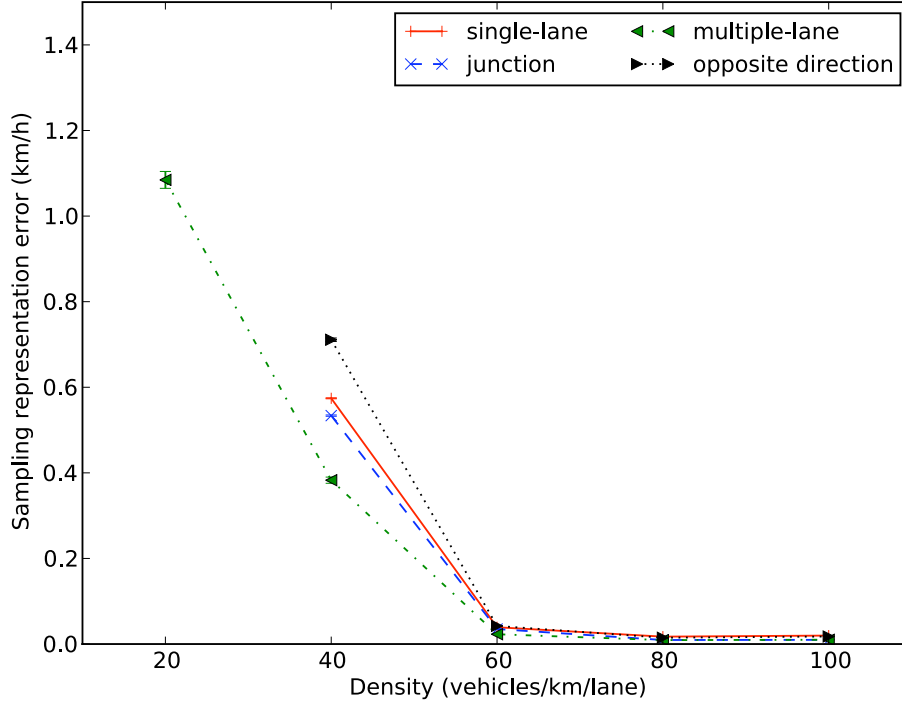


Figure 7.10: Accuracy x density

Evidently, the sampling error depends directly on the parameters defined for the *sensitivity* ϵ function described in Section 5.3.2. However, the results shown here are of high importance, since they show that whenever the TrafficMap information flow has reached the last vehicle located at the opposite end of the road with respect to *flow initiators* with at least one TrafficMap Message, the information has an error limited by 1.1 km/h in low densities scenarios. This may be biased by the specific vehicle distribution used in each scenario and such error is likely to increase whenever it faces a high and frequent speed deviation on the speed profile of vehicles on the road. Nevertheless, the OTHA protocol presented a proper functioning under the unreliable static wireless environment, with the reliance of broadcast messages and the possibility of collisions and errors.

Figure 7.11 depicts a snapshot of the multi-lane road scenario considered in our simulations for one of the lanes at density 20 vehicles/km/lane. The circles represent the entries added by source vehicles, whereas the bars represent vehicles and their speed. We can notice that the entries included in the TrafficMap are able to correctly capture the speed changes present on the road. This figure also illustrates that despite the presence of gaps between vehicles on the road, the end-to-end connectivity could be achieved thanks to the remaining lane of the road.

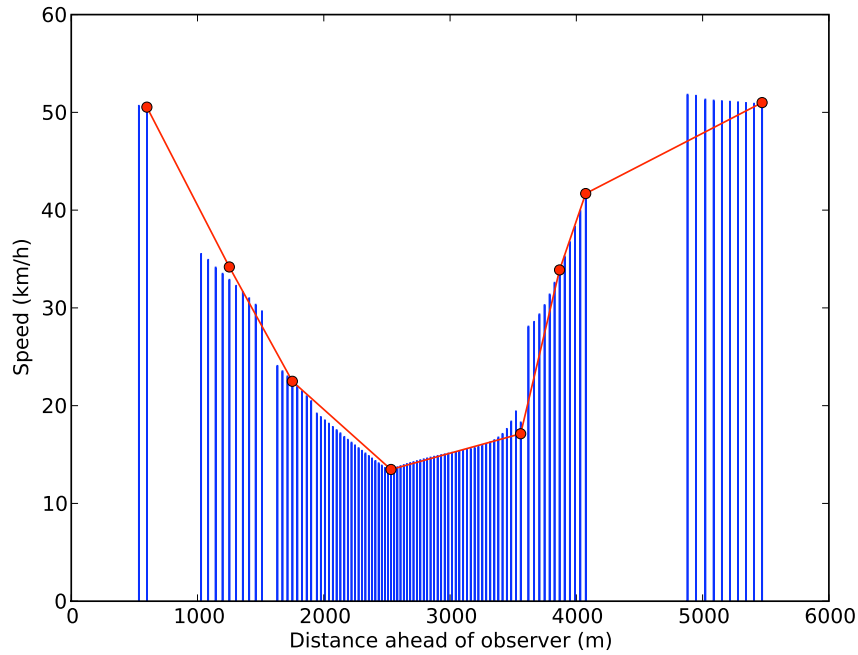


Figure 7.11: Illustration of the matching of TrafficMap entries and the real vehicle trace for one the lanes of the multiple-lane road scenario for a density of 20 vehicles/km/lane

7.3.2.6 Discussion

For almost every metric evaluated the OTHA protocol had its worst performance in the multiple-lane scenario. As we already mentioned, due to a large number of vehicles nearby assigned to a common time slot, the probability of collisions and errors are increased. In fact, the problem is more related to the *time slot density* (vehicles/time slot) rather than the vehicle density on the road (vehicles/km/lane). Because the transmission range of 250 meters assigned to vehicles covers both lanes in the multiple-lane scenario, the number of vehicles is actually the double when compared with a single-lane scenario for the same vehicle density in terms of vehicles/km/lane. This characteristic is particularly severe as we rely on the 802.11p with the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) access mechanism. As we have explained in Section 2.1.1, many challenges exist when relying on the broadcasting of messages with such mechanism: lack of acknowledgments, the hidden terminal problem, and the constant small size for the Contention Window. All these facts together contribute to a great increase in the likelihood of collisions and at the same time decrease the reliability of the protocol in terms of reachability.

In order to cope with such problem, some measures may be taken and evaluated as future work:

- *Employment of power control mechanisms*: the idea is to regulate the transmission power utilized by vehicles in such a manner the number of vehicles within the same range and consequently time slot is decreased. Numerous techniques have

been proposed to control the power level employed in VANETs, for instance, in [Artimy 2007, Gozalvez & Sepulcre 2007, Torrent-Moreno *et al.* 2005, Mittag *et al.* 2008, Chigan & Li 2007, Caizzone *et al.* 2005]

- *Inclusion of number of retries:* the reachability of the protocol may be improved if vehicles have another attempt to propagate the TrafficMap information they hold. One way of introducing a retry mechanism is by resending the last TrafficMap Message in case no messages (echoes) have been heard from other vehicles behind within a certain time period after the transmission.
- *Increase the number of time slots:* if the number of time slots are increased, the number of vehicles assigned per time slots will be lower, as each time slot would comprise a smaller geographical area of the transmission range utilized.

7.4 Mobility Scenarios

7.4.1 Scenario description

The mobility scenario considered in this work merges variations of all previous static scenarios evaluated into a single one, as illustrated in Figure 7.12: a multiple-lane (2 lanes) road R1 with opposite directions, where the lanes of one of the directions are split into two individual lanes by means of a junction point, resulting in the separation of roads R1 and R2. In order to ease the understanding of the results we describe in the following sections, the scenario is divided into Sections 1, 2, 3 and 4. Based on the same notation used for static scenarios the arrow outside the road indicates the direction of vehicles on the road, the dark circles represent flow initiator vehicles and the arrow next to it indicates the message direction of the TrafficMap information flow. Flow initiators are initially chosen as the ones positioned on the extreme side of each road towards the direction of vehicles at time instance zero. In this scenario a single vehicle, represented with a white circle placed in the extreme opposite end of the road when compared with the flow initiator in road R1, is used to gather TrafficMap structures. For the same purpose described for the static scenarios, these TrafficMap structures are used to calculate the accuracy of the information received.

For this scenario, we perform 50 simulation runs, each run with a time duration of 300 seconds (5 minutes). Flow initiators create the TrafficMap structure and start transmitting at time instance zero. Vehicles move at intervals of 0.5 seconds during a simulation run, which yields 600 steps in total. As vehicles move towards their direction they may reach the end of the road. In this case, they are disabled, i.e., have their transmissions canceled and disabled, and are put in a separate place far enough from the scenario described so they cannot disturb the ongoing simulation. Similarly, vehicles may arrive at the opposite end on the road, i.e., the beginning of a road. These vehicles are disabled at the beginning of the simulation and enabled when the simulation duration reaches the time scheduled for them to enter the road. Differently from the simulations performed in static scenarios, now all internal events defined in the upper Traffic Filter Protocol Layer concerning change in speed, lane or road, may be executed whenever a change in these values are detected. Therefore, we evaluate in this section the complete functioning of the OTHA protocol. Moreover, as time evolves different and/or

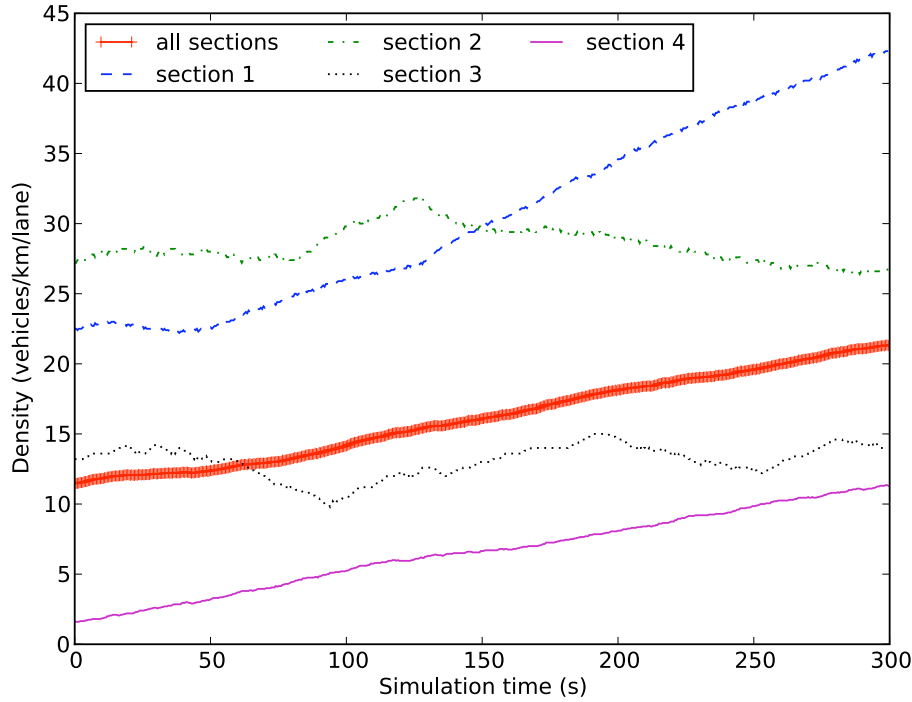


Figure 7.13: Illustration of the mobility scenario: density x time

confidence intervals, this speed profile is a trend during the complete simulation. In addition to the high generation rate of vehicles at the beginning of Section 1, the traffic jam is also caused by the lane change of vehicles as they approach the junction area, e.g., vehicles moving from lane 1 to 2 in R1 to take R2.

The speed behavior in R2 (Figure 7.15) starts with a sudden increase of speed right after vehicles enter the road and it is soon followed by a smooth decrease in speed from 90 km/h down to 77 km/h.

In Section 4 (Figure 7.16), vehicles are generated from the highest position 10 km and travel down to position 0. Their speed begins with similar values around 103 km/h and diverges between the different lanes, i.e., vehicles in lane 1 slowing down to 100 km/h and vehicles in lane 2 speeding up to 110 km/h.

In order to understand the presence of gaps and thus lower multi-hop connectivity among vehicles, we illustrate in Figure 7.17 the maximum theoretical distance of awareness a vehicle positioned at the beginning of each direction of road R1 can expect to receive for each time step of the simulation. We estimate this value by analyzing the maximum multi-hop distance these vehicles can achieve when considering a theoretical transmission range of 250 meters. Based on this estimate, it is clear that Section 4 presents a very sparse network during almost the whole simulation. On the other hand, from Section 1 to 2, a vehicle positioned at the beginning of R1 is able to expect information from other vehicles positioned from 8 km up to 10 km ahead on the road. Similarly to Section 4, the theoretical distance of awareness in Section 3 also fluctuates along time, as a vehicle at the beginning of R1 can only receive information a few times from R2

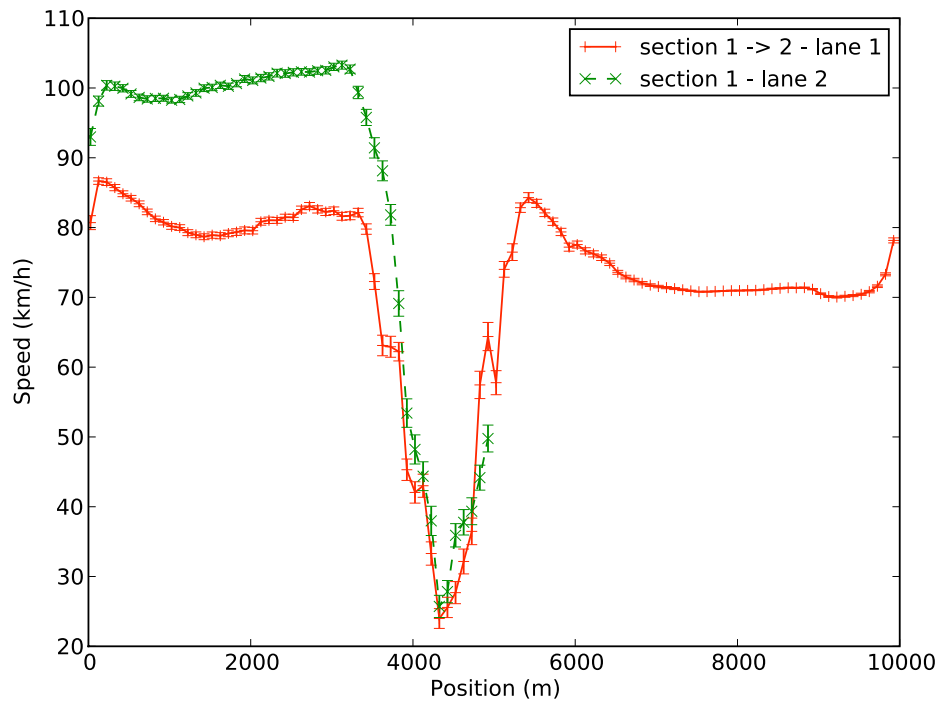


Figure 7.14: Illustration of the mobility scenario: speed x position (Section 1 -> 2)

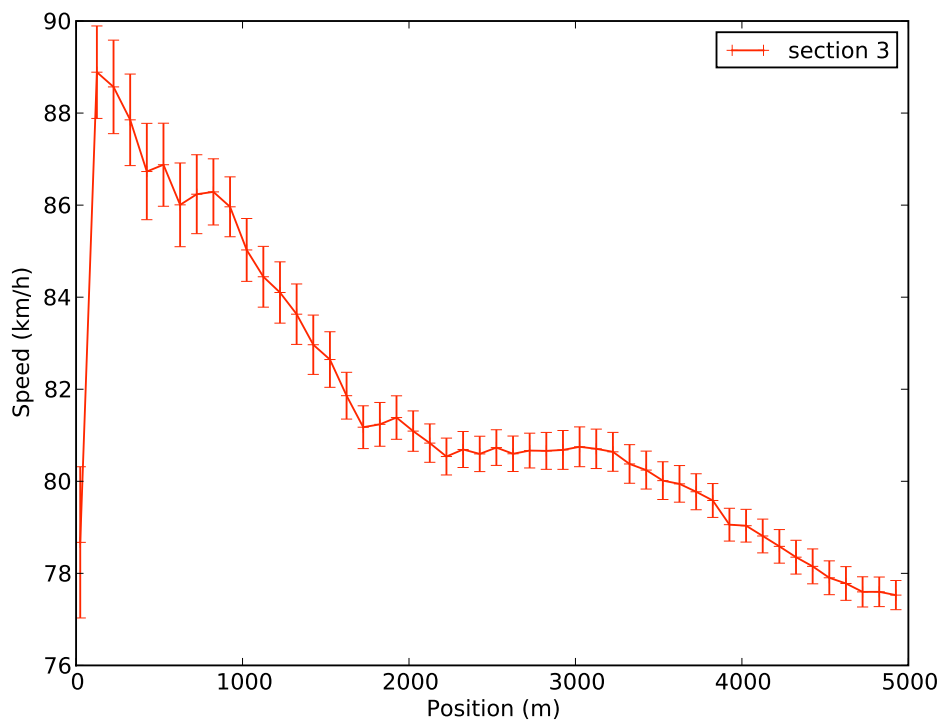


Figure 7.15: Illustration of the mobility scenario: speed x position (Section 3)

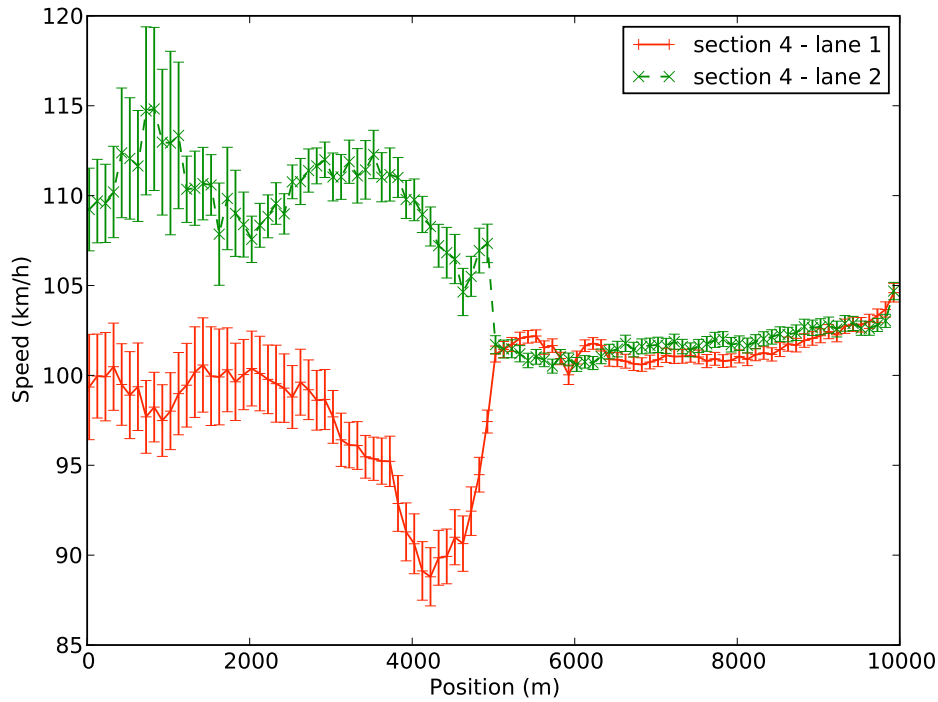


Figure 7.16: Illustration of the mobility scenario: speed x position (Section 4)

during the simulation. This can be verified by observing the distance achieved after the junction point at position 5 km, where R2 begins.

Because we are interested in evaluating the OTHA protocol under scenarios with high speed deviations, preferably with traffic jams to test accuracy, and in a connected network, we concentrate our measurements on roads R1 and R2 and assess the maximum distance of awareness and accuracy metrics. We leave the opposite direction in R1 as a constant background noise that may interfere with ongoing transmissions in R1 and R2. In addition, because in mobility scenarios all internal events in the Traffic Filter Protocol Layer may be executed, there may be more TrafficMap Messages generated during the simulation due to a change of lane or road. For this reason, we also are interested in knowing how these events affect the overall system load. From the metrics described to evaluate the system load, the channel utilization and overhead metrics are evaluated. The remaining metrics, namely, number of receptions and transmission, have been used in the static scenarios to see how they evolve with density. Since the mobility trace utilized in this work presents mainly low densities, their values would not contain much valuable information and are not considered. The remaining metrics, slot utilization, reachability and delay, have already been evaluated under static scenarios and are also not considered. In particular, the maximum distance of awareness achieved for different time instances implicitly tell us something about reachability: the distance of awareness achieved implies that the traffic information has been successfully propagated by means of multi-hop communication and thus by relying on other vehicles in the network (high reachability). Similarly, the accuracy measured includes errors caused by the mobility of vehicles. Thus, a high accuracy achieved means that

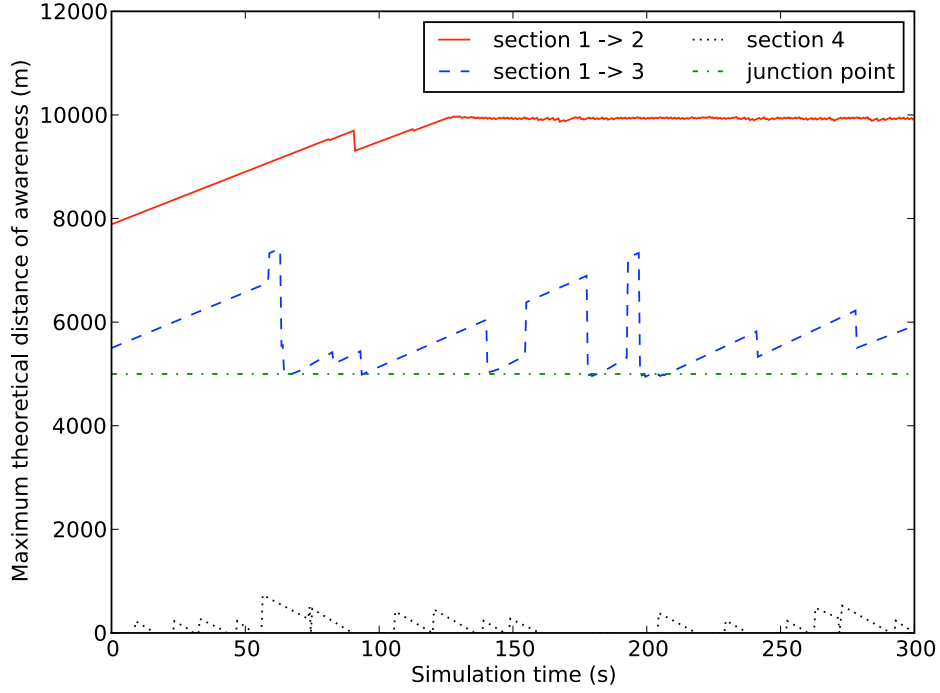


Figure 7.17: Maximum theoretical distance of awareness x time

the delay has been sufficiently low that did not compromise the freshness of the information, which would otherwise affect directly the accuracy.

7.4.2 Results

In this section we present the results for the mobility scenario considered in this work. Similarly to what has been provided in the static scenarios section, we place an example of the TrafficMap output generated during our simulations in Appendix B.2.

7.4.2.1 Distance of awareness

The distance of awareness achieved as time evolves is depicted in Figure 7.18. This graph illustrates the distance of awareness achieved placed over the maximum theoretical distance of awareness illustrated previously in Figure 7.17 for the sections of interest, i.e., Sections 1, 2, and 3. The values sampled from the simulations are the distance averages with 95% confidence intervals for all TrafficMaps received at each time interval of 3 seconds (the MIT interval set to our experiments) for all simulation runs.

From this figure we are able to conclude that the distance of awareness achieved in R1 (from Section 1 to 2) is in great part near the maximum theoretical distance achievable. From Sections 1 to 3, which includes R2 after the junction point, the results demonstrate some fluctuation and displacement especially at the beginning at the simulation. The results with regard to the latter sections are somewhat expected for the following reasons: (i) the maximum theoretical distance

of awareness is calculated for each time instant and it does not take into account the end-to-end delay needed for the complete propagation of the TrafficMap information. The time instance at which a TrafficMap is received may refer to an existing end-to-end connectivity a few seconds before, thus, it may be shifted to the right in this figure; (ii) the constant low density in Section 3 (below 15 vehicles/km/h) results in a sparse network and therefore in lower probability of TrafficMaps to be successfully propagated completely in roads R2; (iii) the maximum theoretical distance of awareness has been estimated with a theoretical transmission range of 250 meters. Even though a proper transmission power has been employed by vehicles to achieve such range, the more distant vehicles are from each other, e.g., near the transmission range limit, the lower is the probability of successful communication.

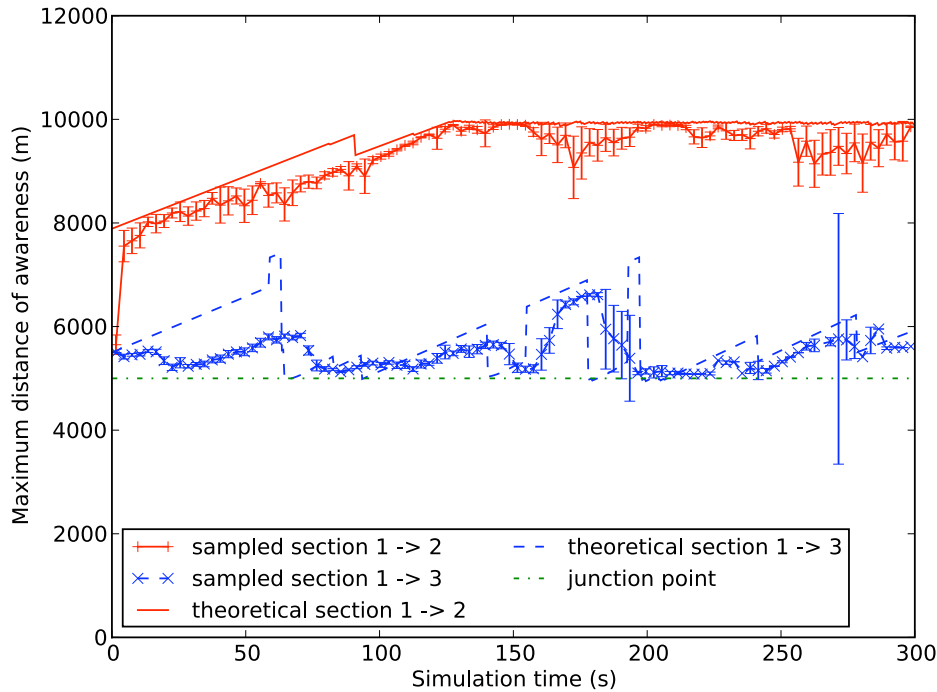


Figure 7.18: Maximum distance of awareness achieved x time

Overall, the results indicate a proper functioning of the OTHA protocol with regard to the distance of awareness achieved. In particular, despite the existing difficulties from Section 1 to 3, the protocol could still provide some awareness to vehicles driving on R1. In addition, the fact that a high distance of awareness has been achieved for different time instances serves as indication that a high reachability has also been achieved. It is expected, however, that under dense networks (over 80 vehicles/km/lane) the connectivity and consequently the distance of awareness may be compromised for the reasons with regard to multiple-lane scenarios presented in Section 7.3.2.6.

7.4.2.2 Accuracy

The accuracy of the traffic information received by means of TrafficMap structures has its evaluation presented in Figure 7.19. For all TrafficMap structures received in all simulation runs the sampling representation error is illustrated in a boxplot. Moreover, the sampling errors are calculated for the time instances at which TrafficMaps have been received. That means that entries within a TrafficMap that have been included priorly, e.g., at the time of TrafficMap construction, may be outdated when compared to the road situation at the moment the TrafficMap has been received. As motivated in Section 7.1.1, this results in an additional error introduced by the mobility of vehicles.

Overall, the results shown in this figure indicate the sampling representation error being concentrated around 5 km/h. 50% of the results are placed between 3.5 and 5.5 km/h. We do not include outliers in this boxplot for the simple reason of making the figure clearer. In this work, outliers are the errors lying more than $1.5 \times IQR$ (interquartile range) lower than the first quartile or $1.5 \times IQR$ higher than the third quartile. Nevertheless, over 95% of the error values fall in between the lower and upper whiskers. The median is found to be around 4.5 km/h and the confidence interval that illustrates its variation (represented by the boxplot notches [McGill *et al.* 1978]) is almost negligible.

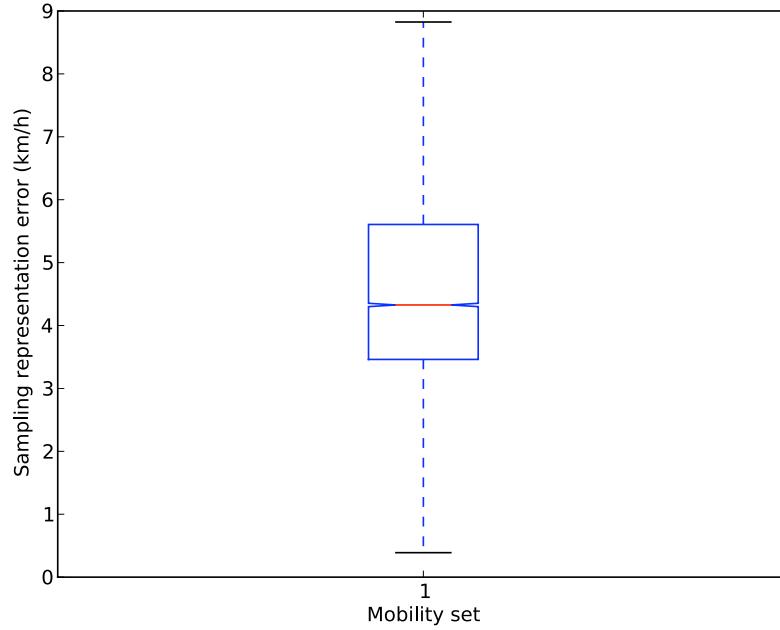


Figure 7.19: Boxplot illustrating the accuracy

Considering the high speed variation of vehicles at some points in the mobility scenario described, for instance the rapid drop in speed (from 100 down to 25 km/h) around position 4500 meters illustrated in Figure 7.14, a sampling error around 5 km/h may be considered low. Furthermore, the speed deviations illustrated in the mentioned figure represent the average behavior in these road sections. For example, Figure 7.20 provides a snapshot at time instance

150 seconds of Section 1 to 2. In this figure we can verify that at individual time instances, the speed deviations may be much higher and so may be the representation error. Note that despite the speed deviations presented in this snapshot, the OTHA protocol was still able to capture the essential existing speed deviations of the referred road sections. We can also verify that the vehicle distribution utilized provides time instances with a much higher speed deviation when compared to the static scenarios previously evaluated in this chapter. Therefore, a higher sampling representation error is somewhat expected.

One can conclude from the above numbers and the given snapshot that the OTHA protocol is able to provide high accuracy in the representation contained in TrafficMaps under mobility scenarios with high speed deviations. Since the accuracy is directly influenced by the thresholds defined in the sensitivity ε function in the upper Traffic Filter Layer, even lower sampling error values can be expected when assigning different (more sensitive) values for these parameters. Furthermore, high accuracy also indicates that the end-to-end delay needed for the propagation of the TrafficMap structure has been sufficiently low in order not to increase the sampling error considerably.

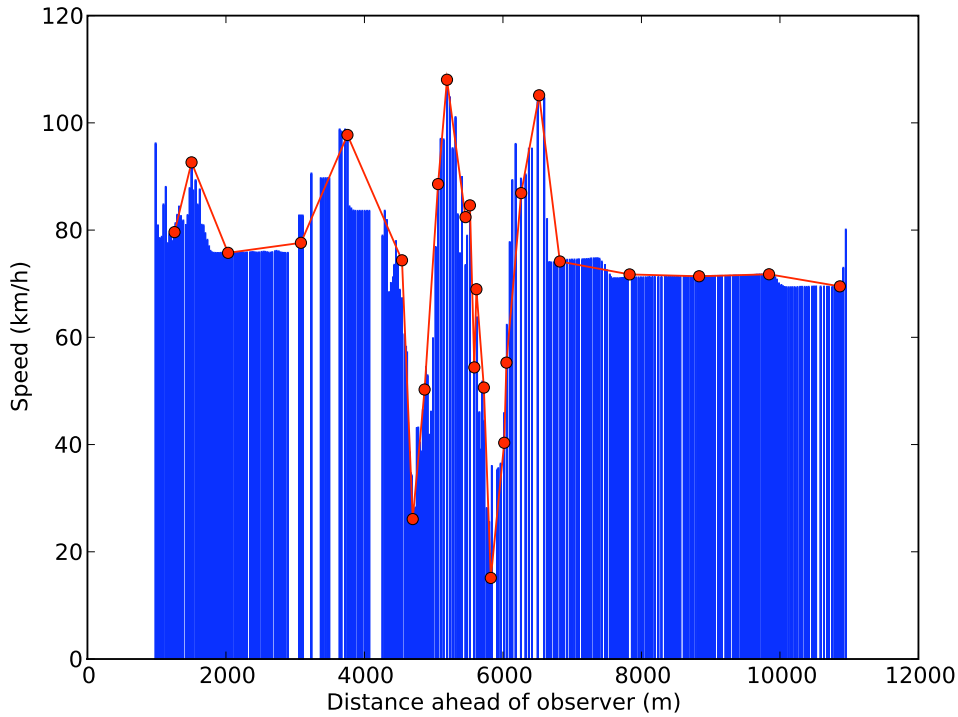


Figure 7.20: Illustration of the matching of TrafficMap entries and the real vehicle trace for lane 1 of Section 1

7.4.2.3 System load

For the mobility scenario considered, the system load is evaluated by means of the total channel utilization and overhead metrics. In all simulation runs, the total channel utilization is measured

by considering the complete period of a simulation period, i.e., 300 seconds. The results for both metrics are represented by boxplots.

Figure 7.21(a) illustrates the overall total channel utilization. 50% of the values are concentrated between 5.2% and 5.7% of the simulation period. The skewness present in the graph indicates that more values fall into lower values with respect to the median. This can be verified by the longer tail towards the lower whisker and also by the median being positioned above the center of the interquartile range. There are no outliers in the data, therefore, 100% of the values are within 4.5% and 6.0% (between lower and upper whiskers). The variation represented by the confidence interval around the median varies from 5.2% to 5.7%. The channel utilization was therefore in worst case 6.0%.

As expected the results indicate a higher total channel utilization in mobility scenarios when compared with the previous static scenarios, where the channel utilization remained less than 0.7% of the total time period assigned to the MIT value. This increase can be explained by the occurrence of interval events which are triggered whenever vehicles change lane or road and their speed differ from the last entry added to the lane/road they moved to. Another possibility is when a speed deviation has been detected by the speed check timer. Due to the occurrence of these events, more TrafficMap Messages requests are sent down to the Dissemination Layer. Consequently, more messages are sent and received by each vehicle, what increases the time spent in utilizing the radio channel. Despite such increase, 6.0% may still give other applications the chance to utilize the same radio channel, as in 94% of the total simulation time vehicles were idle.

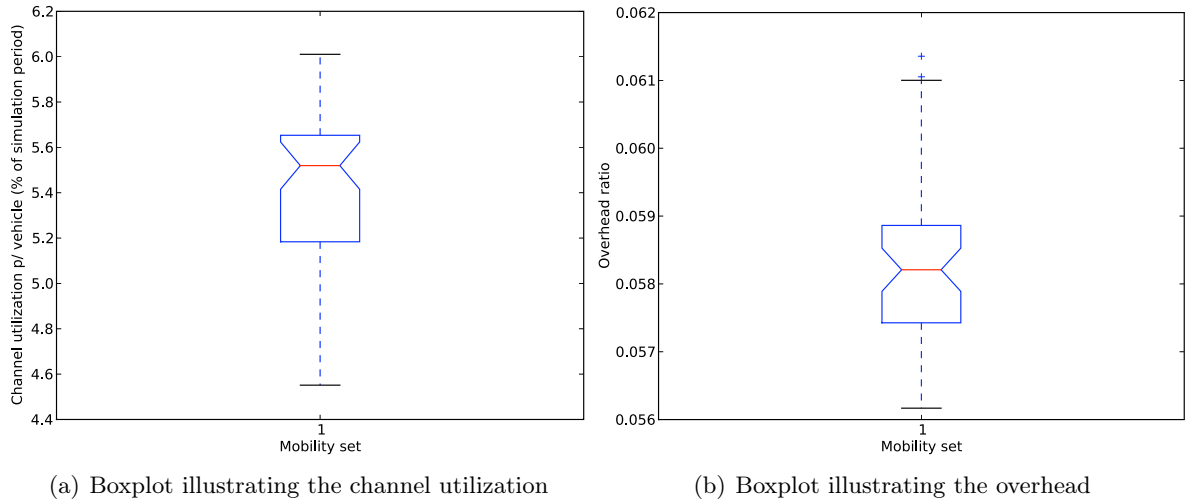


Figure 7.21: Evaluation of the system load

The overhead is depicted in Figure 7.21(b). The upper and lower whiskers include almost 100% of the complete data with exception of two outliers above the upper whisker. Because of the long tail towards the upper whisker and the centralized median with regard to the interquartile range, we can conclude that most values fall above the median. Even when including outliers, the maximum value is below 0.062. This is in fact, close to what has been achieved for static

scenarios, what indicates that the number of receptions is considerably higher than the number of transmissions. The results strengthen the proper functioning of the OTHA protocol, since it is expected that vehicles assigned to different time slots will refrain from broadcasting by the employed suppression technique as they hear an echo of the scheduled message. Thus, more receptions over transmissions are expected.

7.5 Optimization Methods

In this section we present the evaluation results for the optimization methods proposed in each protocol layer, namely, the compression Greedy approach (Section 5.4) in the Traffic Filter Protocol Layer and the time slot improvement (Section 6.4) in the Dissemination Layer.

7.5.1 Compression

Since the compression method to be evaluated has been designed to compress data added to different lanes on the road, we obviously need a multiple-lane scenario. We could, therefore, reutilize both static and mobility scenarios previously described in this chapter. However, as illustrated in Figures 7.14 and 7.16 the multiple-lane sections available in the mobility scenario are either too sparse (Section 4) or present an excessive different speed profile between the lanes (Section 1). Due to these reasons, we limit ourselves to evaluate our compression method, i.e., the Greedy approach, only under the different densities described for the static multiple-lane scenarios. This study will, therefore, serve to gather preliminary results to verify whether or not the Greedy approach in fact may function properly in more realistic scenarios.

Figure 7.22 illustrates the percentage of compression achieved for the different densities considered. As it can be observed, the higher is the density the higher is the percentage of compression achieved. The reasoning behind this pattern is that as the density increases vehicles move gradually more slowly due to congestion, which increases the probability that both lanes have a similar speed profile. At the highest density considered, i.e., 100 vehicles/km/h, the maximum percentage of compression was 35%. Given that the maximum theoretical compression is limited by 50% (one lane being completely compressed to the other), 35% in fact represents a high level of compression.

In terms of numbers, Figure 7.23 shows the evolution of the TrafficMap size along with higher densities. As of density 40, we can notice a large separation between the previous results obtained without compression and the results obtained with compression. While the TrafficMap size of the uncompressed data remains steady around 140 bytes due to a new entry every 1 km defined by the parameter d threshold in the sensitivity ε function, the compressed data size presents an almost linear decrease with density, as the different lanes behave more similarly in terms of speed profile.

In addition to high compression, the compressed data must still be able to accurately represent the current situation of the traffic ahead on the road. For this purpose, Figure 7.24 illustrates the previous accuracy achieved for the uncompressed data together with the compressed data produced by the Greedy approach. Despite the overall increase in sampling error for the multiple-lane compressed, the difference between both results may be considered negligible as in worst case at density 20 the increase in the representation error has been from 1.1

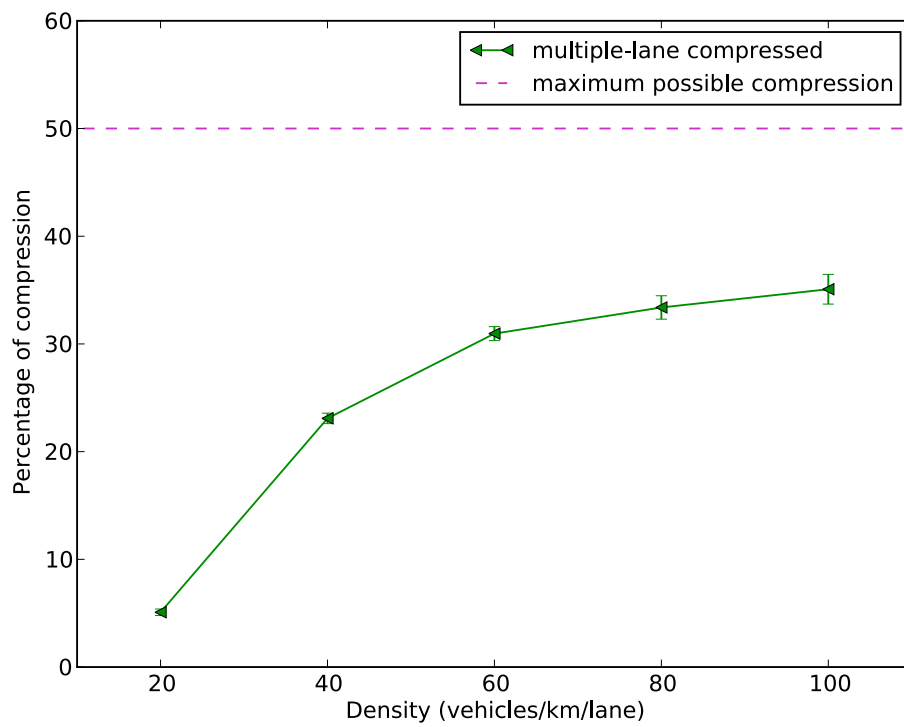


Figure 7.22: Percentage of compression x density

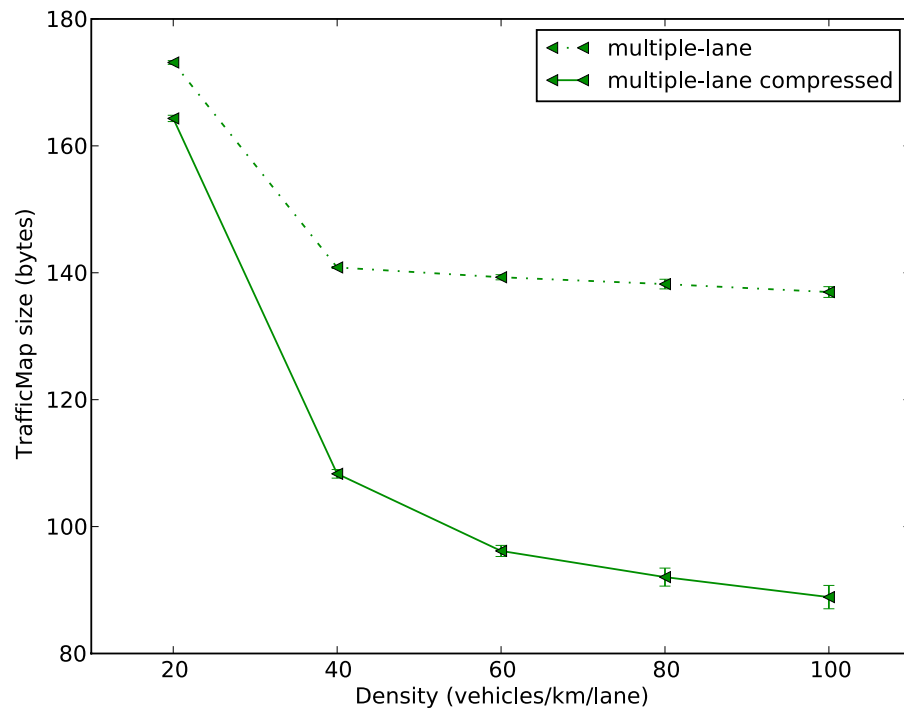


Figure 7.23: TrafficMap size x density

km/h up to 1.5 km/h. Since we rely on the parameters defined in the sensitivity ε function to decide whether or not two entries are similar, a change in these parameters may still improve and consequently diminish the difference presented between the compressed and uncompressed data.

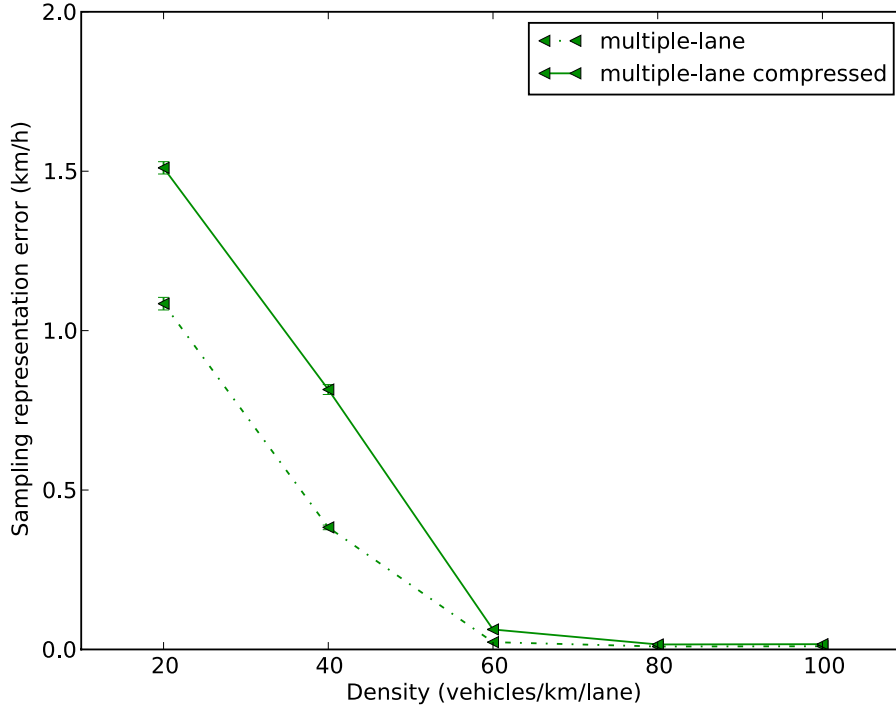


Figure 7.24: Accuracy x density

An important conclusion we may derive from the results above is that despite the evaluation of the compression method only for static scenarios, it is expected that similar results will be possible in congested areas, i.e., traffic jams, where vehicles do not move whatsoever or only at slow speeds. Furthermore, the performance of the Greedy approach in highly congested environments also have shown to be the most accurate due to the high similarity in speed profiles between the lanes. The percentage of compression may still be higher in highways with more than two lanes, as explained in Section 5.4.

7.5.2 Time Slot

The time slot optimization proposed in Section 6.4 aims at improving the time slot utilization in the sense that a higher propagation efficiency, i.e., lower delay, may be achieved. Ideally, each vehicle would have its own estimate number of time slots reserved for relay and source vehicles in a way it is tailored for each area on the road. Therefore, the desired result is a higher utilization of early over later time slots. It is worth remembering that the proposed optimization rely on gathering neighboring information at every time interval defined by the MIT parameter (3 seconds in our simulation configuration) in order to provide such estimate. The underlying

assumption is that the current traffic behavior, i.e., whether there are speed deviations on the road, does not change as frequently as the MIT period. Because our goal is to validate this assumption, the optimization method is only evaluated in mobility scenarios, as static scenarios.

Figure 7.25(a) illustrates a comparison between the time slot utilization in our mobility scenario with the normal configuration (without the time slot optimization) with the optimization method being employed. The results show that, differently from expected, generally later time slots have been utilized with time slot optimization, which results in a less efficient propagation, i.e., a longer delay.

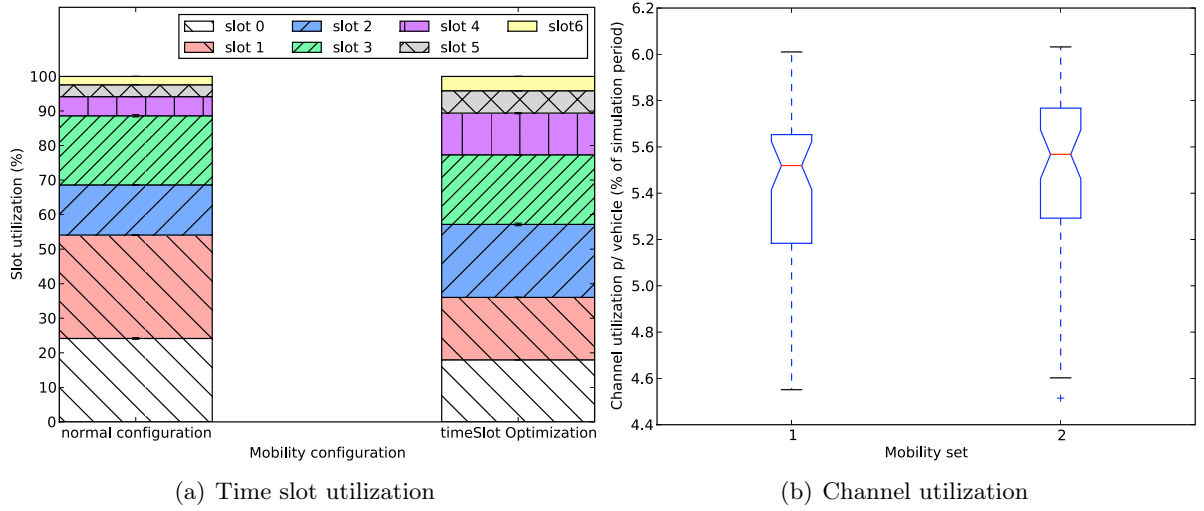


Figure 7.25: Evaluation of the time slot optimization method

In order to confirm the results above we verify whether the channel load has been increased or not with the presence of the time slot optimization method. Figure 7.25(b) illustrates by means of boxplots that in fact the channel load has been slightly higher with the optimization method being employed, what confirms that the time slots have not been well distributed among relay and source vehicles. The consequence is that more transmissions and hence receptions have been carried out by vehicles.

The poor performance achieved by the time slot optimization method may be explained by two reasons: (i) the assumption that the speed profiles on the road remain similar after 3 seconds is false; (ii) the approach of distributing the total number of time slots to source and relay vehicles simply based on the proportion of source and relay neighbor vehicles is not adequate. Further study on both aspects is required for further conclusions.

Conclusion

Contents

8.1 General Conclusions	123
8.2 Answers to Research Questions	124
8.3 Future Work	126

In this chapter we present the conclusions of this work and identify topics for future work.

The chapter is further structured as follows: Section 8.1 provides general conclusions of this work; Section 8.2 presents answers for the research questions raised at the beginning of this work; and finally, Section 8.3 discusses future work.

8.1 General Conclusions

In this work, we have proposed a networking solution to attend the information requirements of the Congestion Assistant in highway real-world scenarios. The information required about upcoming traffic jams is provided by means of speed profiles regarding each lane of roads ahead. In particular, the following Highway Real-World Scenarios have been considered: *single-lane roads*, *multiple-lane roads*, *junctions*, and *roads with multiple (opposite) directions*. Our solution has been translated into a communication protocol, the OTHA protocol, that altogether comprises extensions and modifications to the communication protocol presented in [van Eenennaam 2008].

Within a vehicle, Intelligent Transportation Systems (ITS) have been structured with two layers: the Application and Communication layers. The OTHA protocol relies on information envisioned to be obtained from a Navigation System which, together with the Congestion Assistant, belongs to the upper Application Layer. The lower Communication Layer comprises the network protocol layers. The OTHA protocol is deployed on the Communication Layer and runs above the MAC and PHY layers defined by the IEEE 802.11p standard.

The OTHA protocol functions as the network and application network protocol layers in the OSI reference model for telecommunications and as such it has been organized in two layers: (i) the Traffic Filter Protocol Layer and (ii) the Dissemination Protocol Layer. In (i) we have concentrated our efforts in providing an accurate view of the upcoming traffic by utilizing the minimum possible amount of information. This traffic information has been stored in a structure referred to as the TrafficMap. In order to achieve such goal, we have presented functions to select only crucial characteristic of the road as well as compression mechanisms to reduce the size of existing entries at the cost of some loss of information. Differently, in (ii) our goal has been on efficiently disseminating the TrafficMap data among vehicles by broadcasting TrafficMap

Messages by means of multi-hop vehicle-to-vehicle communication. Our dissemination strategy is based on the broadcast suppression technique presented in [Wisitpongphan *et al.* 2007] with additional changes to include the different vehicle types considered, namely, source and relay vehicles.

The performance of the OTHA protocol has been evaluated by means of simulation. For this purpose, we have utilized OMNET++ with the Mobility Framework. Various metrics have been used to evaluate both layers of the protocol and their proposed optimization methods. We assessed the protocol both in a controlled environment with static scenarios and under a more realistic scenario consisting of vehicle traces with high mobility and speed variations. The results obtained indicate good performance of the protocol with respect to the metrics evaluated. In particular, the performance of the protocol in multiple-lane scenarios has been deteriorated with high vehicle densities. Furthermore, from the optimization methods proposed, namely, the Greedy approach and the Time Slot optimization, the former has presented promising results with regard to the level of compression achieved and low loss in accuracy whereas the latter has not contributed with the improvement of the time slot utilization.

Overall, the protocol presented a proper function in the real-world scenarios considered and may serve as a starting point for the development of a system meant to provide an over-the-horizon awareness to vehicles on the road, not only to the Congestion Assistant, but to other applications that may require such information.

8.2 Answers to Research Questions

As raised during the introduction of this work, the following questions can now be answered:

- ***Single-lane highways:*** how to address more realistic scenarios in a manner the solution does not compromise the overall performance in presented in [van Eenennaam 2008] for simple single-lane roads?

By means of simulations, we have shown that the OTHA protocol is capable of addressing more realistic scenarios and still presenting a desirable performance in terms of reachability, delay, system load, accuracy, and time slot utilization. In fact, single-lane roads are the basis for the construction of more complex scenarios, such as multiple-lane roads or single-lane roads with junctions and, therefore, a proper functioning in this scenario is essential.

- ***Multiple-lane highways:*** which information must be included from each lane on the road in order to provide an accurate view of the traffic ahead? How to efficiently coordinate the exchange of information in scenarios with vehicles driving nearby in different lanes?

In order to provide a precise speed profile of each lane in a multiple-lane scenario, the lane number of vehicles has been utilized. Together with speed and position values, the lane number is used to individually characterize the traffic behavior in each lane on the road.

The Dissemination Protocol Layer has been designed to provide an efficient exchange of information by means of a directional broadcast of TrafficMap Messages. Great effort has been put into providing high accurate information by giving source vehicles higher priority

in the communication. Moreover, in order to prevent that a large number of messages is utilized simultaneously and thus increase the system load, different timers have been used to establish an upper and lower bound for the period a vehicle must wait before transmitting a message.

The simulation results for multiple-lane scenarios have indicated that the OTHA protocol functions properly with regard to the metrics considered for both static and mobility scenarios. However, a deterioration in performance of the OTHA protocol has been verified under high vehicle density scenarios. This is mainly due to a higher number of vehicles scheduled to transmit in a common time slot defined by our suppression technique in the Dissemination Layer and the consequent high probability of collisions. In order to cope with this issue, several improvements have been proposed and left as future work as we describe in Section 8.3.

- ***Highways with vehicles moving in different directions:* which are the impacts of having vehicles from multiple directions utilizing a vehicle-to-vehicle communication networking solution? Which information and how to efficiently coordinate its exchange among vehicles in order to provide an accurate view of the traffic?**

In scenarios with roads with multiple directions, it is crucial that the exact source location of the information received is identified. Therefore, in addition to the lane number, the road identification and direction of vehicles are included.

Based on the simulation results obtained in this work, the presence of independent TrafficMap flows traveling in different directions has not affected whatsoever the performance of the protocol in either static or mobility scenarios. However, further evaluation is required to verify whether this result is valid in more general scenarios, as the lack of coordination among multiple flows may result in more transmission errors and collisions.

- ***Junctions in highways:* which information must be included from each road linked to a junction in order to provide an accurate view of the traffic ahead? How could the exchange of information be coordinated in such scenarios where traffic information may be originated by vehicles from multiple roads?**

In addition to the exact source location of the information received, vehicles may benefit from the use of the type of junction points and their location. By utilizing this information, we can derive which are the possible roads a vehicle can take in the upcoming traffic and provide only the relevant information to vehicles behind.

The evaluation performed in both static and mobility scenarios have indicated good performance of the OTHA protocol in every aspect considered under the presence of road junctions. In particular, the merging function defined in the upper Traffic Filter Protocol Layer has played the important role of gathering multiple TrafficMap flows coming from different roads and merging them into a single TrafficMap structure and thus into a single TrafficMap flow. Interestingly, the FFP timer defined in the Dissemination Protocol Layer has contributed with the merging function by preventing vehicles from transmitting

multiple TrafficMap Messages in sequence, which gave multiple flows the opportunity to be merged.

8.3 Future Work

During the development of this work, several opportunities of improvement have been raised but no accomplished due to time constraints:

- In the sensitivity ε function, vehicles are considered as source vehicles whenever a new entry is added to the TrafficMap. In particular, the d threshold determines the maximum distance a certain lane in the road segment can remain without a refresh of information from vehicles driving on it. Because this information may not be crucial for the correct representation of the road, whenever a new entry is added for this reason, vehicles could be simply considered as relay instead of source vehicles. This could diminish the amount of information propagated in the network, since messages sent by relay vehicles rely on previous message IDs and, therefore, can be canceled.
- As mentioned in Section 6.4, the total number of time slots used by the suppression mechanism may be adjusted to the current traffic density of vehicles. This would increase the propagation efficiency of TrafficMap Messages, as vehicles may be better distributed among the time slots.
- In this work, for the sake of simplification, whenever vehicles received TrafficMap Messages from the opposite direction of the road, they would simply discard them. However, the opposite direction may be used to relay information from both directions and thus increase the reachability of the protocol, especially in sparse networks. A possible approach is to consider sparse networks as *Delay Tolerant Networks (DTNs)*, as described in [Franck & Gil-Castineira 2007]. Such networks often use a *store-carry-forward* communication model that relies on the mobility of nodes to transfer messages when nodes are geographically separated.

In addition, as outlined in Section 7.3.2.6, the following measures may be taken to cope with the deterioration of performance found in multiple-lane scenarios:

- *Employment of power control mechanisms*: the idea is to regulate the transmission power utilized by vehicles in such a manner the number of vehicles within the same range and consequently time slot is decreased. Numerous techniques have been proposed to control the power level employed in VANETs, for instance, in [Artimy 2007, Gozalvez & Sepulcre 2007, Torrent-Moreno *et al.* 2005, Mittag *et al.* 2008, Chigan & Li 2007, Caizzone *et al.* 2005]
- *Inclusion of number of retries*: the reachability of the protocol may be improved if vehicles have another attempt to propagate the TrafficMap information they hold. One way of introducing a retry mechanism is by resending the last TrafficMap Message in case no messages (echoes) have been heard from other vehicles behind within a certain time period after the transmission.

-
- *Increase the number of time slots:* if the number of time slots are increased, the number of vehicles assigned per time slots will be lower, as each time slot would comprise a smaller geographical area of the transmission range utilized.

Illustration of Static Scenarios

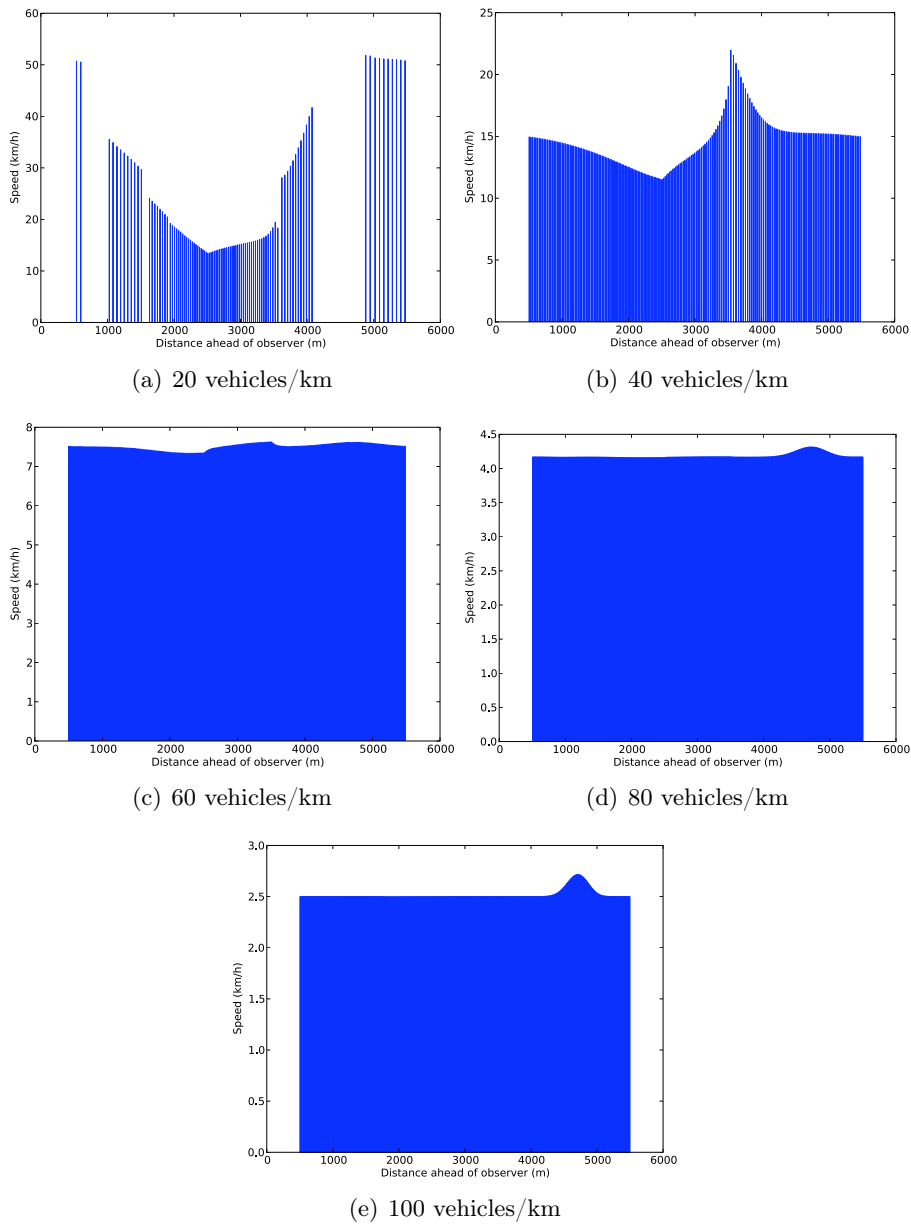


Figure A.1: Single-lane

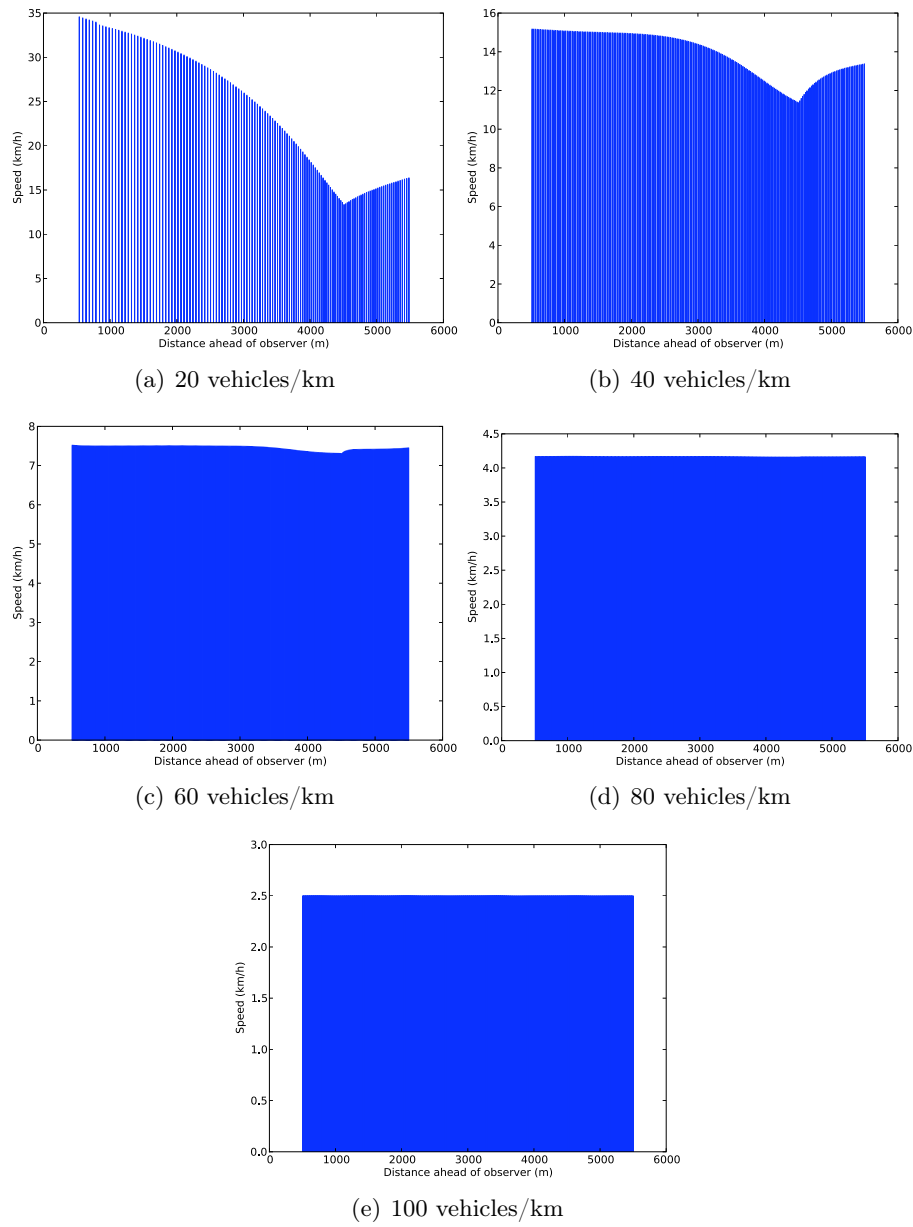


Figure A.2: Multiple-lane - Lane 1

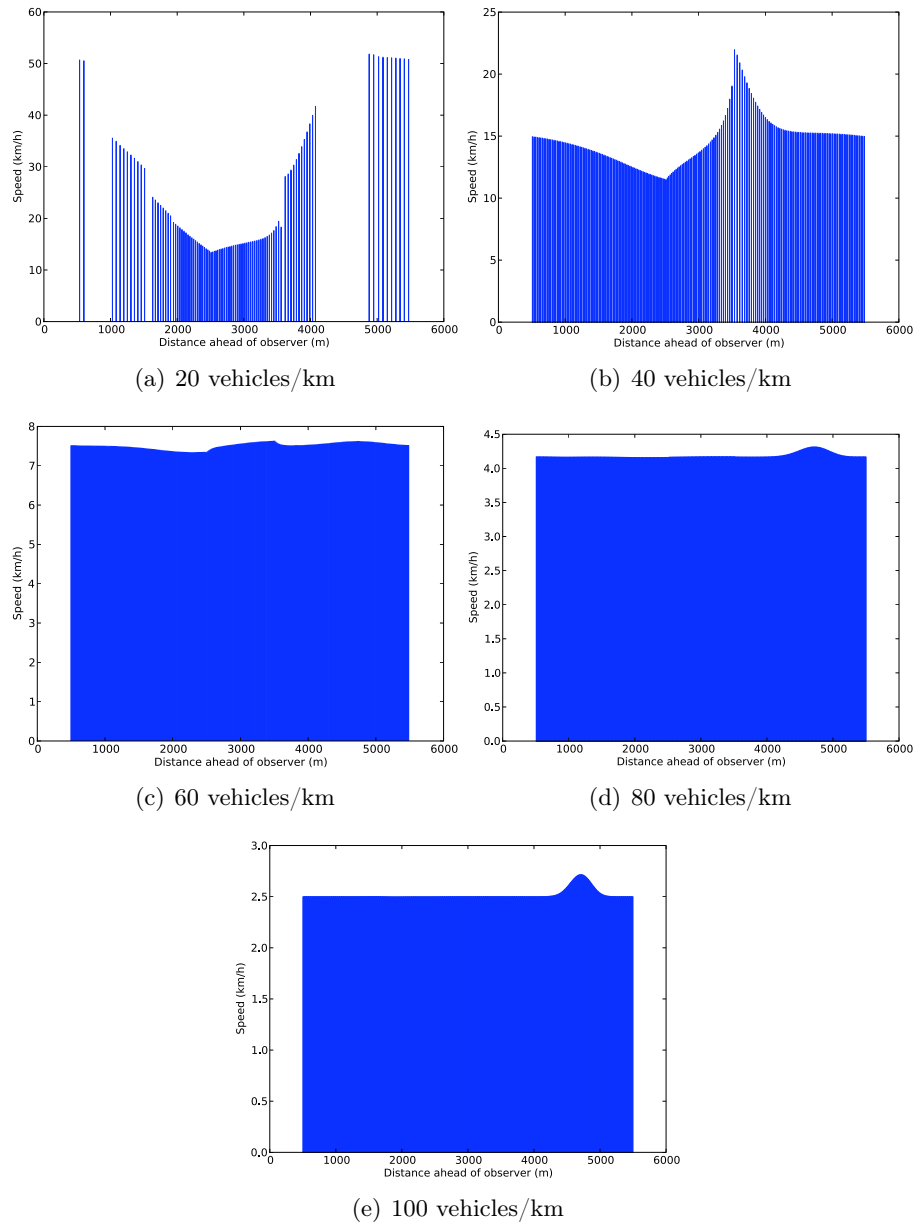


Figure A.3: Multiple-lane - Lane 2

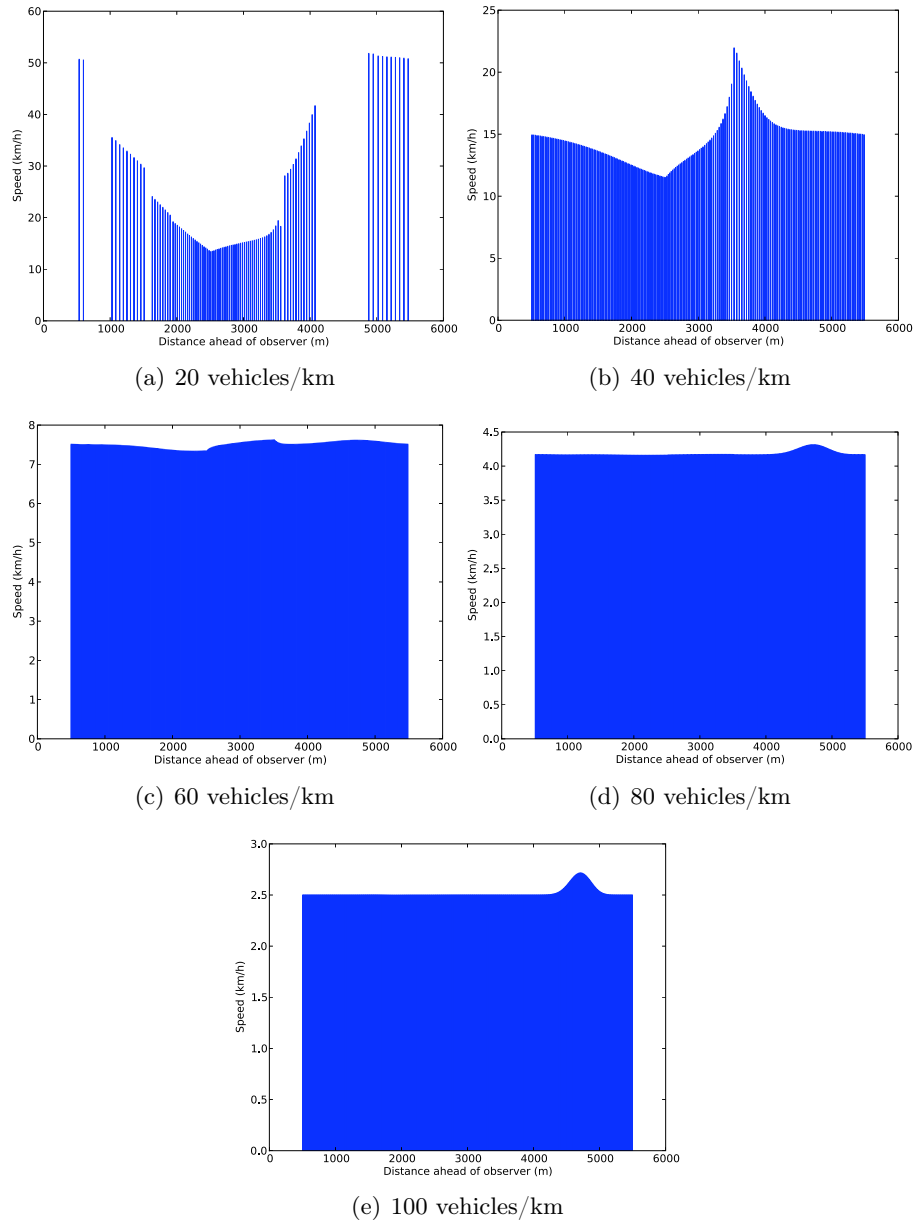


Figure A.4: Junction - Road 1

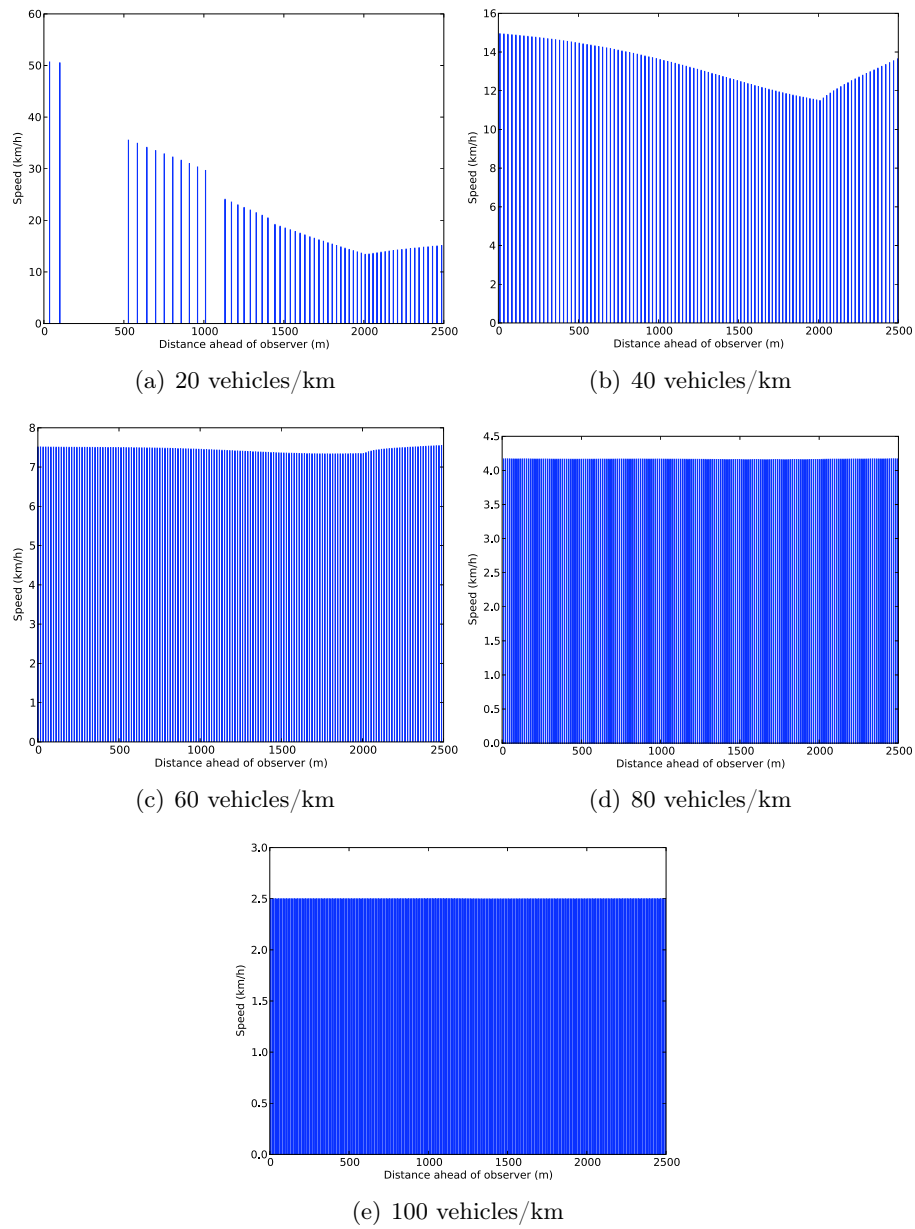


Figure A.5: Junction - Road 2

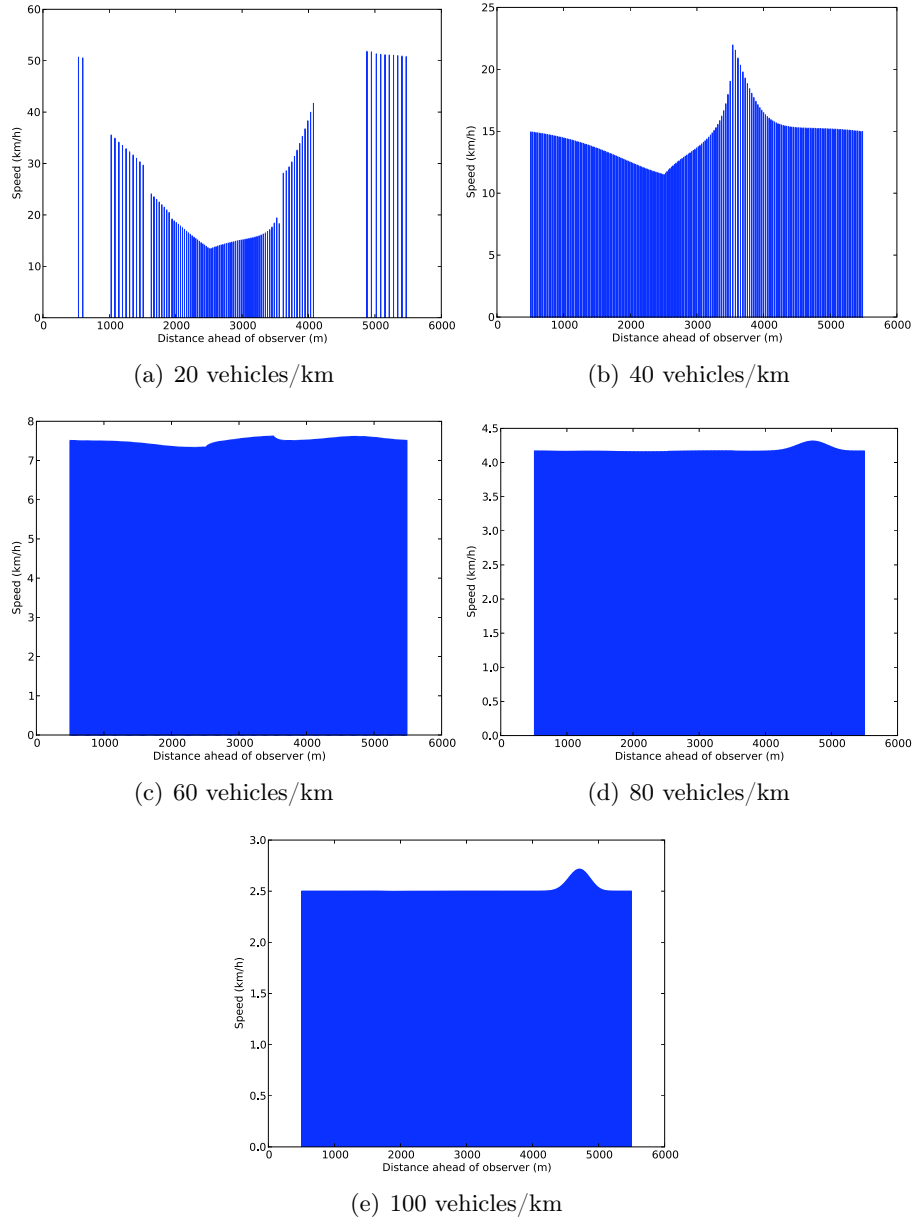


Figure A.6: Opposite direction - Direction 1

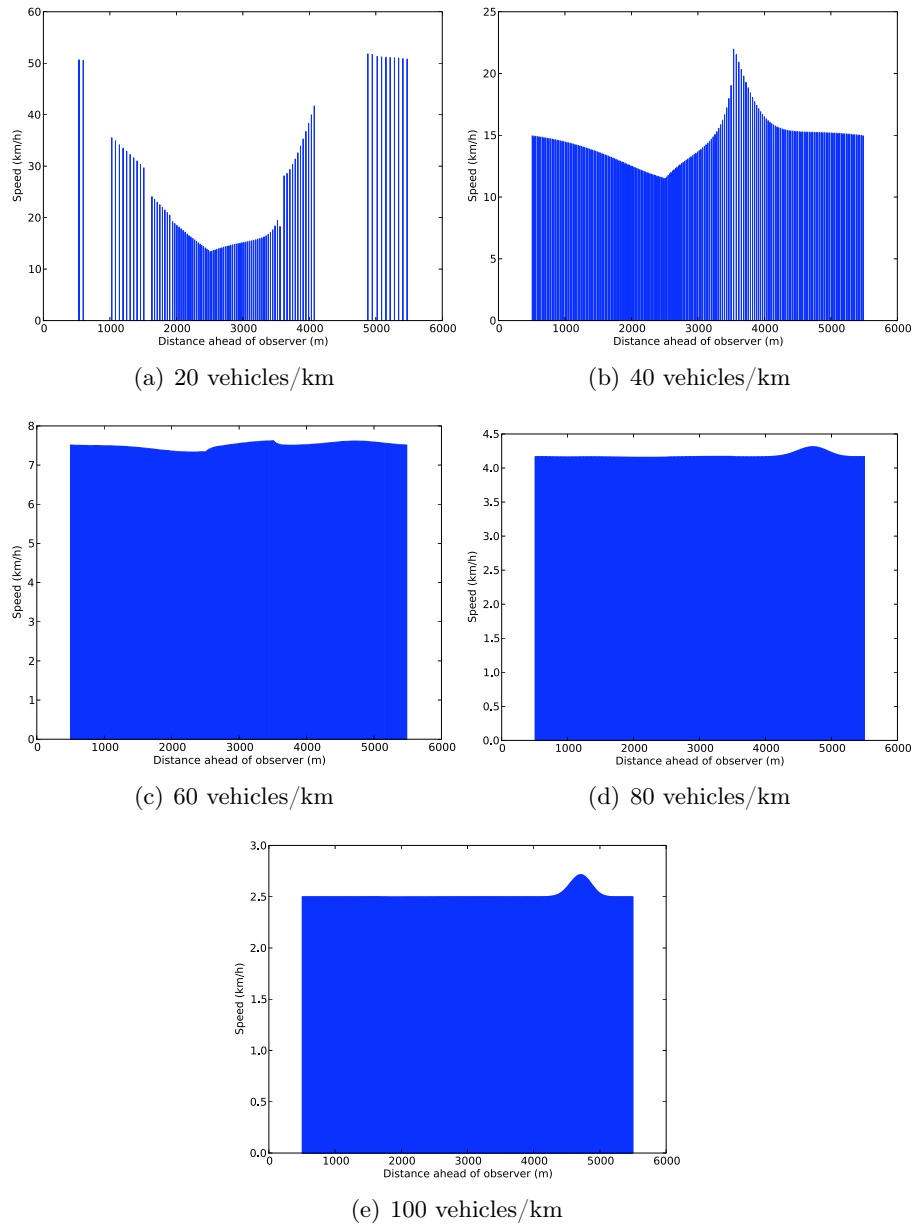


Figure A.7: Opposite direction - Direction 2

TrafficMap Entries

B.1 Static Scenario

B.1.1 Single-lane

Node: 0

RoadID: 0 Direction.x: 0 Direction.y: 500 LaneNr: 1

PosX: 5484 PosY: 500 Speed: 15.031 timeStamp: 0

PosX: 4474 PosY: 500 Speed: 15.534 timeStamp: 0.075981259902

PosX: 3465 PosY: 500 Speed: 15.899 timeStamp: 0.162234682775

PosX: 2447 PosY: 500 Speed: 11.9843 timeStamp: 0.248082172265

PosX: 1440 PosY: 500 Speed: 14.0392 timeStamp: 0.334288303501

B.1.2 Multiple-lane

Node: 0

RoadID: 0 Direction.x: 0 Direction.y: 500 LaneNr: 1

PosX: 5482 PosY: 500 Speed: 16.3697 timeStamp: 0

PosX: 4460 PosY: 500 Speed: 14.5443 timeStamp: 0.078126798402

PosX: 3325 PosY: 500 Speed: 23.8982 timeStamp: 0.160640157647

PosX: 2287 PosY: 500 Speed: 29.7754 timeStamp: 0.25007809767

PosX: 1272 PosY: 500 Speed: 32.88 timeStamp: 0.339658458993

RoadID: 0 Direction.x: 0 Direction.y: 500 LaneNr: 2

PosX: 5471 PosY: 504 Speed: 50.9976 timeStamp: 0.000796666666

PosX: 4074 PosY: 504 Speed: 41.6955 timeStamp: 0.108312756704

PosX: 3867 PosY: 504 Speed: 33.8812 timeStamp: 0.11953156142

PosX: 3559 PosY: 504 Speed: 17.1266 timeStamp: 0.149833003067

PosX: 2532 PosY: 504 Speed: 13.4722 timeStamp: 0.219330385237

PosX: 1751 PosY: 504 Speed: 22.4904 timeStamp: 0.281033455424

PosX: 1252 PosY: 504 Speed: 34.1862 timeStamp: 0.339658458993

PosX: 599 PosY: 504 Speed: 50.533 timeStamp: 0.39072451697

B.1.3 Junction

Node: 0

RoadID: 0 Direction.x: 0 Direction.y: 500 LaneNr: 1
PosX: 5484 PosY: 500 Speed: 15.0327 timeStamp: 9
PosX: 4474 PosY: 500 Speed: 15.5577 timeStamp: 9.075700155657
PosX: 3465 PosY: 500 Speed: 16.3899 timeStamp: 9.170750232738
PosX: 2447 PosY: 500 Speed: 11.8415 timeStamp: 9.366444668321
PosX: 1440 PosY: 500 Speed: 14.0448 timeStamp: 9.519887057884

RoadID: 1 Direction.x: 0 Direction.y: 500 LaneNr: 1
PosX: 4264.23 PosY: 2264.23 Speed: 13.1401 timeStamp: 9
PosX: 3555.71 PosY: 1555.71 Speed: 12.9686 timeStamp: 9.075486605695
PosX: 2839.41 PosY: 839.411 Speed: 14.6715 timeStamp: 9.170699211889

B.1.4 Opposite direction

Node: 0

RoadID: 0 Direction.x: 0 Direction.y: 500 LaneNr: 1
PosX: 5484 PosY: 500 Speed: 15.0357 timeStamp: 3
PosX: 4474 PosY: 500 Speed: 15.5006 timeStamp: 3.075548568316
PosX: 3465 PosY: 500 Speed: 16.3899 timeStamp: 3.171791058056
PosX: 2447 PosY: 500 Speed: 11.8885 timeStamp: 3.297113571717
PosX: 1440 PosY: 500 Speed: 14.0744 timeStamp: 3.402389665572

Node: 397

RoadID: 0 Direction.x: 12000 Direction.y: 500 LaneNr: 1
PosX: 509 PosY: 520 Speed: 14.8371 timeStamp: 3
PosX: 1534 PosY: 520 Speed: 13.1902 timeStamp: 3.084598851562
PosX: 2552 PosY: 520 Speed: 12.6321 timeStamp: 3.171003116033
PosX: 3537 PosY: 520 Speed: 20.3773 timeStamp: 3.257145788845
PosX: 4550 PosY: 520 Speed: 15.2384 timeStamp: 3.353467470041

B.2 Mobility Scenario

Node: 1139

RoadID: 0 Direction.x: 0 Direction.y: 500 LaneNr: 1
PosX: 9773.84 PosY: 502 Speed: 69.483 timeStamp: 66.59947021957

PosX: 8759.19 PosY: 502 Speed: 71.967 timeStamp: 66.721976012406
PosX: 7738.55 PosY: 502 Speed: 71.4113 timeStamp: 66.807484530885
PosX: 6734.88 PosY: 502 Speed: 81.3869 timeStamp: 67.023202208263
PosX: 6174.06 PosY: 502 Speed: 66.3043 timeStamp: 67.072403907239
PosX: 5908.64 PosY: 505.25 Speed: 51.44 timeStamp: 67.083790957823
PosX: 5859.61 PosY: 505.25 Speed: 69.7002 timeStamp: 67.252503841615
PosX: 5582.92 PosY: 505.25 Speed: 94.44 timeStamp: 67.263747787954
PosX: 5368.84 PosY: 505.25 Speed: 66.25 timeStamp: 67.318957421811
PosX: 5265.64 PosY: 505.25 Speed: 25.8856 timeStamp: 67.346903908729
PosX: 5047.8 PosY: 505.25 Speed: 16.65 timeStamp: 67.502945587838
PosX: 5036.5 PosY: 502.45 Speed: 7.57 timeStamp: 67.503777629535
PosX: 5011.67 PosY: 502.45 Speed: 30.16 timeStamp: 67.553194231842
PosX: 4872.35 PosY: 505.25 Speed: 82.31 timeStamp: 67.55975373353
PosX: 3891.35 PosY: 505.25 Speed: 102.11 timeStamp: 67.711882540573
PosX: 3334.52 PosY: 505.25 Speed: 77.4939 timeStamp: 67.744969807288
PosX: 2696.71 PosY: 505.25 Speed: 94.7828 timeStamp: 67.81494933565
PosX: 2165.43 PosY: 505.25 Speed: 88.6192 timeStamp: 68.021834208813
PosX: 1164.33 PosY: 505.25 Speed: 78.4638 timeStamp: 68.140437957128

RoadID: 0 Direction.x: 0 Direction.y: 500 LaneNr: 2

PosX: 5952.67 PosY: 501.75 Speed: 55.42 timeStamp: 67.083229728485
PosX: 5925.18 PosY: 504.9 Speed: 45.18 timeStamp: 67.094686466585
PosX: 5817.3 PosY: 501.75 Speed: 68.91 timeStamp: 67.195306441666
PosX: 5749.95 PosY: 501.75 Speed: 97.547 timeStamp: 67.196627771178
PosX: 5366.56 PosY: 501.75 Speed: 65.56 timeStamp: 67.218339421812
PosX: 5270.8 PosY: 501.75 Speed: 31.46 timeStamp: 67.230341747253
PosX: 5134.4 PosY: 501.75 Speed: 1.8 timeStamp: 67.24018193946
PosX: 5023.38 PosY: 501.75 Speed: 27.39 timeStamp: 67.251472685701
PosX: 4752.13 PosY: 501.75 Speed: 104.949 timeStamp: 67.514298530108
PosX: 3752.08 PosY: 501.75 Speed: 109.601 timeStamp: 67.613552819998
PosX: 2750.35 PosY: 501.75 Speed: 108.93 timeStamp: 67.712360872157
PosX: 1714.56 PosY: 501.75 Speed: 104.542 timeStamp: 68.069857411735

RoadID: 1 Direction.x: 0 Direction.y: 500 LaneNr: 1

PosX: 6602.04 PosY: 620.13 Speed: 97.6131 timeStamp: 54.011129467064
PosX: 6355.97 PosY: 571.86 Speed: 87.0045 timeStamp: 60.002047633357

Bibliography

- [Akyildiz *et al.* 2002] IF Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. *Wireless sensor networks: a survey*. Computer networks, vol. 38, no. 4, pages 393–422, 2002. 29
- [Artimy 2007] M. Artimy. *Local Density Estimation and Dynamic Transmission-Range Assignment in Vehicular Ad Hoc Networks*. vol. 8, no. 3, pages 400–412, 2007. 36, 83, 108, 126
- [Ayanoglu & Sabnani 1997] E. Ayanoglu and K.K. Sabnani. *Navigation system for an automotive vehicle*, November 18 1997. US Patent 5,689,252. 21
- [Baldessari *et al.* 2007] R. Baldessari, B. Bodekker, M. Deegener, A. Festag, W. Franz, C. Kellum, T. Kosch, A. Kovacs, M. Lenardi, C. Meniget *al.* *Car-2-Car Communication Consortium-Manifesto (Version 1.1)*. In IEEE Vehicular Technology Conference, 2007. 28, 30
- [Blum & Eskandarian 2009] J.J. Blum and A. Eskandarian. *Avoiding Timeslot Boundary Synchronization for Multihop Message Broadcast in Vehicular Networks*. 2009 IEEE 69th Vehicular Technology Conference: VTC2009-Spring 26–29 April 2009, Hilton Diagonal Mar Hotel, Barcelona, Spain, 2009. 73, 74
- [Caizzzone *et al.* 2005] G. Caizzzone, P. Giacomazzi, L. Musumeci and G. Verticale. *A power control algorithm with high channel availability for vehicular ad hoc networks*. In Proc. IEEE International Conference on Communications ICC 2005, volume 5, pages 3171–3176, 16–20 May 2005. 108, 126
- [Chausse *et al.* 2005] F. Chausse, J. Laneurit and R. Chapuis. *Vehicle localization on a digital map using particles filtering*. In IEEE Intelligent Vehicles Symposium, 2005. Proceedings, pages 243–248, 2005. 22
- [Chigan & Li 2007] Chunxiao Chigan and Jialiang Li. *A Delay-Bounded Dynamic Interactive Power Control Algorithm for VANETs*. In Proc. IEEE International Conference on Communications ICC '07, pages 5849–5855, 24–28 June 2007. 108, 126
- [Cormen *et al.* 2001] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein. Introduction to algorithms. MIT press, 2001. 52
- [Cozart 2000] Justin Cozart. *Northwest at the junction of the Dallas North Tollway - <http://www.flickr.com/photos/fatguyinalittlecoat/2890078135/>*, January 2000. ix, 17
- [Danescu *et al.* 2007] R. Danescu, S. Nedevschi, MM Meinecke and TB To. *Lane Geometry Estimation in Urban Environments Using a Stereovision System*. In IEEE Intelligent Transportation Systems Conference, 2007. ITSC 2007, pages 271–276, 2007. 22

- [Davis *et al.* 1996] C. Davis, P. Vixie, T. Goodwin and I. Dickinson. *RFC 1876: A means for expressing location information in the domain name system*, Jan. 1996. Updates RFC1034, RFC1035 [135,136]. Status: Experimental, 1996. 51, 94
- [Drytkiewicz *et al.* 2003] W. Drytkiewicz, S. Sroka, V. Handziski, A. Koepke and H. Karl. *A mobility framework for omnet++*. In 3rd International OMNeT++ Workshop, 2003. 93
- [Farkas *et al.* 1997] D. Farkas, J. Young, B. Baertlein and U. Ozguner. *Forward-looking radar navigation system for 1997 AHS demonstration*. In IEEE Conference on Intelligent Transportation System, 1997. ITSC'97., pages 672–675, 1997. 22
- [Flow 2009] INRIX Dynamic Predictive Flow. <http://www.inrix.com/predictive.asp>, August 2009. 2
- [Foundation 2009] E. Foundation. *Eclipse*, 2009. 93
- [Franck & Gil-Castineira 2007] L. Franck and F. Gil-Castineira. *Using Delay Tolerant Networks for Car2Car communications*. In IEEE International Symposium on Industrial Electronics, 2007. ISIE 2007, pages 2573–2578, 2007. 126
- [Friis 1946] HT Friis. *A note on a simple transmission formula*. Proceedings of the IRE, vol. 34, no. 5, pages 254–256, 1946. 96
- [Gast & Loukides 2002] M.S. Gast and M. Loukides. 802.11 wireless networks: the definitive guide. O'Reilly & Associates, Inc. Sebastopol, CA, USA, 2002. 5, 29, 94
- [Gozalvez & Sepulcre 2007] J. Gozalvez and M. Sepulcre. *Opportunistic-Driven Adaptive Radio Resource Management Technique for Efficient Wireless Vehicular Communications*. In Proc. VTC-2007 Fall Vehicular Technology Conference 2007 IEEE 66th, pages 2116–2120, Sept. 30 2007–Oct. 3 2007. 108, 126
- [Hartenstein & Laberteaux 2008] H. Hartenstein and K. Laberteaux. *A Tutorial Survey on Vehicular Ad Hoc Networks*. IEEE Communications Magazine, vol. 46, no. 6, 2008. 2
- [Herman & Prigogine 1979] Robert Herman and Ilya Prigogine. *A Two-Fluid Approach to Town Traffic*. Science, vol. 204, no. 4389, pages 148–151, 1979. 36
- [Hofmann-Wellenhof *et al.* 2001] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins. Gps theory and practice. New York, NY, 2001. 21
- [Hurst 1995] SR Hurst. *The characteristic function of the Student t distribution*. Res. Rep. FMRR, vol. 6, page 95, 1995. 87
- [IEEE 2006] IEEE. *IEEE Draft Amendment to Standard for Information Technology - Telecommunications and Information exchange between systems - Local and Metropolitan networks - specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment: Wireless Access in Vehicular Environments*. IEEE 802.11p Draft 2.0, November 2006. 9, 26, 96

- [IEEE 2009] IEEE. *Intelligent Transportation Systems* - <http://www.ewh.ieee.org/tc/its/>, March 2009. 11
- [ISO 2002] ISO. *Intelligent transport systems - Geographic Data Files (GDF) - Overall data specification*. Rapport technique Revision of ISO/TR 14825:1996, International Organization for Standardization, 2002. 23
- [Jiang & Delgrossi 2008] D. Jiang and L. Delgrossi. *IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments*. In Proc. IEEE Vehicular Technology Conference VTC Spring 2008, pages 2036–2040, 11–14 May 2008. 9, 96
- [Johannson 1999] H. Johannson. *Impact of Ecodriving on emissions and fuel consumption, a pre-study*. Swedish National Road Administration report 1999: 165E, 1999. i, 1
- [KatyFreeway 2009] KatyFreeway. *Texas Department of Transportation - Katy Freeway* - <http://www.katyfreeway.org/>, March 2009. 15, 35
- [Kerner & Rehborn 1996] BS Kerner and H. Rehborn. *Experimental features and characteristics of traffic jams*. Physical Review E, vol. 53, no. 2, pages 1297–1300, 1996. 71
- [Kim et al. 2007] S. Kim, J.H. Park, S.I. Cho, S. Park, K. Lee and K. Choi. *Robust Lane Detection for Video-Based Navigation Systems*. In 19th IEEE International Conference on Tools with Artificial Intelligence, 2007. ICTAI 2007, volume 2, 2007. 22
- [Live 2009] TomTom GO 740 Live. *GPSReview* - <http://www.gpsreview.net/tomtom-740-live/>, August 2009. 2
- [Marsden et al. 2001] G. Marsden, M. McDonald and M. Brackstone. *Towards an understanding of adaptive cruise control*. Transportation Research Part C, vol. 9, no. 1, pages 33–51, 2001. 22
- [McCanne et al. 2000] S. McCanne, S. Floyd, K. Fall, K. Varadhan et al. *Network simulator ns-2*, 2000. 93
- [McGill et al. 1978] R. McGill, J.W. Tukey and W.A. Larsen. *Variations of box plots*. American Statistician, pages 12–16, 1978. 115
- [Mittag et al. 2008] Jens Mittag, Felix Schmidt-Eisenlohr, Moritz Killat, Jérôme Härri and Hannes Hartenstein. *Analysis and design of effective and low-overhead transmission power control for VANETs*. In VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking, pages 39–48, New York, NY, USA, 2008. ACM. 35, 58, 108, 126
- [Motorists 2007] UK Motorists. *The Highway Code* - <http://www.ukmotorists.com/>, September 2007. 15
- [Nagel & Schreckenberg 1992] K. Nagel and M. Schreckenberg. *A cellular automaton model for freeway traffic*. J. Phys. I France, vol. 2, pages 2221–2229, 1992. 36

- [NAVTEQ 2009] NAVTEQ. <http://www.navteq.com/>, July 2009. 23
- [Ni *et al.* 1999] S.Y. Ni, Y.C. Tseng, Y.S. Chen and J.P. Sheu. *The broadcast storm problem in a mobile ad hoc network*. In Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pages 151–162. ACM New York, NY, USA, 1999. 63, 64
- [Orfanus *et al.* 2008] D. Orfanus, J. Lessmann, P. Janacik and L. Lachev. *Performance of wireless network simulators: a case study*. In Proceedings of the 3rd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, pages 59–66. ACM New York, NY, USA, 2008. 93
- [Ortiz 2008] Mariordo Mario Roberto Duran Ortiz. *Rodovia dos Bandeirantes* - http://en.wikipedia.org/wiki/Rodovia_dos_Bandeirantes, June 2008. 16
- [Pipes 1953] Louis A. Pipes. *An Operational Analysis of Traffic Dynamics*. Journal of Applied Physics, vol. 24, no. 3, pages 274–281, 1953. 36
- [Quadstone 2004] L. Quadstone. *Quadstone Paramics v5. 0 Modeller User Guide*. Scotland, UK, 2004. 109
- [Rayanchu *et al.* 2008] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha and S. Banerjee. *Diagnosing wireless packet losses in 802.11: Separating collision from weak signal*. In IEEE INFOCOM 2008. The 27th Conference on Computer Communications, pages 735–743, 2008. 6
- [Rezaei *et al.* 2007] S. Rezaei, R. Sengupta, H. Krishnan and X. Guan. *Reducing the Communication Required By DSRC-Based Vehicle Safety Systems*. In IEEE Intelligent Transportation Systems Conference, 2007. ITSC 2007, pages 361–366, 2007. 35
- [Schiller 2003] J.H. Schiller. *Mobile communications*. Addison Wesley, 2003. ix, 5, 7, 65
- [Schönhof & Helbing 2009] M. Schönhof and D. Helbing. *Criticism of three-phase traffic theory*. Transportation Research Part B, 2009. 36
- [Shibata *et al.* 2006] N. Shibata, T. Terauchi, T. Kitani, K. Yasumoto, M. Ito and T. Higashino. *A method for sharing traffic jam information using inter-vehicle communication*. Proceedings of V2VCOM, 2006. 36
- [Stevens & Rago 2005] R.W. Stevens and S.A. Rago. *Advanced programming in the unix (r) environment*. Addison-Wesley Professional, 2005. 94
- [Suthaputehakun & Ganz 2007] C. Suthaputehakun and A. Ganz. *Priority based inter-vehicle communication in vehicular ad-hoc networks using IEEE 802.11 e*. In Proceedings of the 65th IEEE Vehicular Technology Conference (VTC'07), pages 2595–2599, 2007. 96
- [Taylor *et al.* 2006] G. Taylor, C. Brunsdon, J. Li, A. Olden, D. Steup and M. Winter. *GPS accuracy estimation using map matching techniques: Applied to vehicle positioning and odometer calibration*. Computers, Environment and Urban Systems, vol. 30, no. 6, pages 757–772, 2006. 21

- [Tele-Atlas 2009] Tele-Atlas. <http://www.teleatlas.com/index.htm?Lang=EN>, July 2009. 23
- [Telematique 2009] Telematique. http://telematique.eu/enabling_technologies/gdf.en.html, July 2009. 23
- [TLN 2007] TLN. *Transport in cijfers*. Rapport technique, Transport en Logistiek Nederland, 2007. 1
- [TomTom 2009] TomTom. <http://www.tomtom.com/>, August 2009. 2
- [Tonguz et al. 2007] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige and V. Sadekar. *Broadcasting in VANET*. 2007 Mobile Networking for Vehicular Environments, pages 7–12, 2007. 58
- [Torrent-Moreno et al. 2005] Marc Torrent-Moreno, Paolo Santi and Hannes Hartenstein. *Fair sharing of bandwidth in VANETs*. In VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks, pages 49–58, New York, NY, USA, 2005. ACM. 35, 108, 126
- [Toyota 2009] Toyota. *Safety Toyota* - http://www.safetytoyota.com/en-gb/lane_keeping_assist.html#tab_technology, August 2009. 22
- [Treiber et al. 2000] M. Treiber, A. Hennecke and D. Helbing. *Congested traffic states in empirical observations and microscopic simulations*. Arxiv preprint cond-mat/0002177, 2000. 36, 97
- [Tseng et al. 2001] Y.C. Tseng, S.Y. Ni and E.Y. Shih. *Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network*. In Distributed Computing Systems, 2001. 21st International Conference on., pages 481–488, 2001. 64
- [van Driel 2007] Cornelia Jacoba Geertje van Driel. *Driver support in congestion : an assessment of user needs and impacts on driver and traffic flow*. PhD thesis, Enschede, November 2007. i, 1, 11, 14
- [van Eenennaam 2008] E.M. van Eenennaam. *Providing over-the-horizon awareness to driver support systems by means of multi-hop ad hoc vehicle -to-vehicle communication*, December 2008. i, ix, 2, 3, 4, 11, 12, 13, 14, 19, 25, 31, 36, 37, 42, 45, 46, 49, 67, 68, 71, 73, 74, 76, 89, 91, 93, 94, 96, 123, 124
- [Varga et al. 2001] A. Varga et al. *The OMNeT++ discrete event simulation system*. In Proceedings of the European Simulation Multiconference (ESM'2001), pages 319–324, 2001. 93
- [Victoria 1994] Victoria. *Parliament of Victoria - Road Safety Committee* - <http://www.parliament.vic.gov.au/rsc/DEMERIT/demerit3.htm#6>, November 1994. 22
- [Webster 1978] D.G. Webster. *Speedometer and odometer apparatus*, February 14 1978. US Patent 4,074,196. 22

- [Wetteroth 2001] D. Wetteroth. Osi reference model for telecommunications. McGraw-Hill Professional, 2001. 26
- [Wikipedia 2007] Wikipedia. *Junction (Road)* - [http://en.wikipedia.org/wiki/Junction_\(road\)](http://en.wikipedia.org/wiki/Junction_(road)), May 2007. 17
- [Wischhof 2007] Lars Wischhof. *Self-Organizing Communication in Vehicular Ad Hoc Networks*. PhD thesis, 2007. 35
- [Wisitpongphan *et al.* 2007] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai and V. Sadekar. *Broadcast storm mitigation techniques in vehicular ad hoc networks*. vol. 14, no. 6, pages 84–94, December 2007. ix, 2, 64, 65, 66, 67, 72, 79, 94, 124
- [Witte 1992] F. Witte. *Forward looking radar*, July 21 1992. US Patent 5,132,686. 22
- [Xian *et al.* 2008] X. Xian, W. Shi and H. Huang. *Comparison of OMNET++ and other simulator for WSN simulation*. In 3rd IEEE Conference on Industrial Electronics and Applications, 2008. ICIEA 2008, pages 1439–1443, 2008. 93
- [Xu *et al.* 2002] K. Xu, M. Gerla and S. Bae. *How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks*. In IEEE Global Telecommunications Conference, 2002. GLOBECOM'02, volume 1, 2002. 96
- [Zhang *et al.* 2005] H. Zhang, D. Hou and Z. Zhou. *A Novel Lane Detection Algorithm Based on Support Vector Machine*. In Progress In Electromagnetics Research Symposium, pages 22–26, 2005. 22